Lecture Notes in Computer Science 2687

Edited by G. Goos, J. Hartmanis, and J. van Leeuwen

Springer
Berlin
Heidelberg
New York
Barcelona
Hong Kong
London
Milan
Paris
Tokyo

José Mira José R. Álvarez (Eds.)

Artificial Neural Nets Problem Solving Methods

7th International Work-Conference on
Artificial and Natural Neural Networks, IWANN 2003
Maó, Menorca, Spain, June 3-6, 2003
Proceedings, Part II

Springer

Volume Editors

José Mira
José R. Álvarez
Universidad Nacional de Educación a Distancia
E.T.S. de Ingeniería Informática
Departamento de Inteligencia Artificial
Juan del Rosal, 16, 28040 Madrid, Spain
E-mail:{jmira/jras}@dia.uned.es

Cataloging-in-Publication Data applied for

A catalog record for this book is available from the Library of Congress

Bibliographic information published by Die Deutsche Bibliothek
Die Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliographie;
detailed bibliographic data is available in the Internet at <http://dnb.ddb.de>.

CR Subject Classification (1998): F.1, F.2, I.2, G.2, I.4, I.5, J.3, J.4, J.1

ISSN 0302-9743
ISBN 3-540-40211-X Springer-Verlag Berlin Heidelberg New York

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

http://www.springer.de

Printed in Germany

Typesetting: Camera-ready by author, data conversion by DA-TeX Gerd Blumenstein
Printed on acid-free paper SPIN: 10927830 06/3142 5 4 3 2 1 0

Preface

The global purpose of IWANN conferences has been to provide a broad and interdisciplinary forum for the interplay between neuroscience and computation. Our dream has been, and still is: (1) find ways to understand the physiological, symbolic and cognitive nature of the nervous system (NS) with the help of computational and engineering tools; and (2) find rich and insightful sources of inspiration in biology, to develop new materials, mechanisms and problem-solving methods (PSM) of value in engineering and computation. As all of us know well, this dream started with the Ancient Greeks, reappeared in the foundational stage of neurocybernetics and bionics, and is now broadly accepted in the scientific community under different labels such as computational neuroscience (CN) and artificial neural nets (ANN), or genetic algorithms and hybrid neuro-fuzzy systems.

We have also to recognize that there is a considerable lack of credibility associated with CN and ANN among some researchers, both from biology and from the engineering and computation area. Potential causes of this scepticism could be the lack of methodology, formal tools, and real-world applications, in the engineering area, and the lack also of formal tools for cognitive process modeling. There is also the possibility of the computational paradigm being inappropriate to explain cognition, because of the "representational" character of any computational model. Some "more situated" approaches are looking back to the "neurophysiological epistemology" of the 1960's (mind in a body) to search directly for the mechanisms that could embody the cognitive process, and this means some fresh air for our old dream of connectionism. The tremendous difficulties of these fields (CN and ANN) and of the questions addressed (How does the NS work, and how can we mimic this work in a non-trivial manner?) justify the delay in getting proper answers.

We hope that the papers in these two volumes and the discussions during this seventh working conference will help us to create some new critique and a constructive atmosphere. The neural modeling community and all the people engaged in the connectionist perspective of knowledge engineering would certainly benefit from this new atmosphere.

IWANN 2003, the 7th International Work-Conference in Artificial and Natural Neural Networks took place in Maó, Menorca, Spain, during June 3–6, 2003, addressing the following topics:

Mathematical and computational methods in neural modeling. Levels of analysis. Brain theory. Neural coding. Mathematical biophysics. Population dynamics and statistical modeling. Diffusion processes. Dynamical binding. Synchronization. Resonance. Regulatory mechanisms. Cellular automata.

Neurophysiological data analysis and modeling. Ionic channels. Synapses. Neurons. Circuits. Biophysical simulations.

Structural and functional models of neurons. Analogue, nonlinear, recurrent, RBF, PCA, digital, probabilistic, Bayesian, fuzzy and object-oriented formulations.

Learning and other plasticity phenomena. Supervised, non-supervised, reinforcement and statistical algorithms. Hybrid formulations. Incremental and decremental architectures. Biological mechanisms of adaptation and plasticity. Development and maturing.

Complex systems dynamics. Statistical mechanics. Attractors. Optimization, self-organization and cooperative-competitive networks. Evolutionary and genetic algorithms.

Cognitive Processes and Artificial Intelligence. Perception (visual, auditive, tactile, proprioceptive). Multi-sensory integration. Natural language. Memory. Decision making. Planning. Motor control. Neuroethology. Knowledge modeling. Multi-agent systems. Distributed AI. Social systems.

Methodology for nets design, simulation and implementation. Data analysis, task identification and recursive design. Development environments and editing tools. Implementation. Evolving hardware.

Bio-inspired systems and engineering. Bio-cybernetics and bionics. Signal processing, neural prostheses, retinomorphic systems, and other neural adaptive prosthetic devices. Molecular computing.

Applications. Artificial vision, speech recognition, spatio-temporal planning and scheduling. Data mining. Sources separation. Applications of ANNs in robotics, astrophysics, economics, the Internet, medicine, education and industry.

IWANN 2003 was organized by the Universidad Nacional de Educación a Distancia, UNED, Madrid, in cooperation with IFIP (Working Group in Neural Computer Systems, WG10.6) and the Spanish RIG IEEE Neural Networks Council.

Sponsorship was obtained from the Spanish Ministerio de Ciencia y Tecnología under project TIC2002-10752-E and the organizing university (UNED).

The papers presented here correspond to talks delivered at the conference. After the evaluation process, 197 papers were accepted for oral or poster presentation, according to the recommendations of referees and the author's preferences. We have organized these papers into two volumes arranged basically following the topics list included in the call for papers. The first volume, entitled "Computational Methods in Neural Modeling," is divided into six main parts and includes the contributions on: biological models, functional models, learning, self-organizing systems, artificial intelligence and congnition, and bioinspired developments.

In the second volume, "Artificial Neural Nets Problem-Solving Methods," we have included the contributions dealing with nets design, simulations, implementations, and application developments.

We would like to express our sincere gratitude to the members of the organizing and program committees, in particular to Félix de la Paz López, to the referees, and to the organizers of pre-organized sessions for their invaluable effort

in helping with the preparation of this conference. Thanks also to the invited speakers for their effort in preparing the plenary lectures.

Last, but not least, the editors would like to thank Springer-Verlag, in particular Alfred Hofmann, for the continuous and excellent cooperative collaboration, from the first IWANN in Granada (1991, LNCS 540), to the successive meetings in Sitges (1993, LNCS 686), Torremolinos (1995, LNCS 930), Lanzarote (1997, LNCS 1240), Alicante (1999, LNCS 1606 and 1607), again in Granada (2001, LNCS 2084 and 2085), and now in Maó.

June 2003

José Mira
José R. Álvarez

Scientific Committee (Referees)

Ajith Abraham, Oklahoma State University (USA)
Igor Aizenberg, Neural Networks Technologies Ltd. (Israel)
Igor Aleksander, Imperial College of Science, Technology and Medicine (UK)
Amparo Alonso Betanzos, Universidade da Coruña (Spain)
José Ramón Álvarez Sánchez, UNED (Spain)
Miguel Atencia, Universidad de Málaga (Spain)
Antonio Bahamonde, Universidad de Oviedo en Gijón (Spain)
Emilia I. Barakova, RIKEN (Japan)
Senen Barro Ameneiro, Universidade de Santiago de Compostela (Spain)
Valeriu Beiu, Washington State University (USA)
Gustavo A. Camps i Valls, Universitat de València (Spain)
Andreu Català Mallofré, Universitat Politècnica de Catalunya (Spain)
Carlos Cotta, Universidad de Málaga (Spain)
Félix de la Paz López, UNED (Spain)
Angel P. del Pobil, Universitat Jaume-I (Spain)
Ana E. Delgado Garcia, UNED (Spain)
Jose Dorronsoro, Universidad Autónoma de Madrid (Spain)
Richard Duro, Universidade da Coruña (Spain)
Reinhard Eckhorn, Philips University (Germany)
Marcos Faúndez-Zanuy, Universitat Politècnica de Catalunya (Spain)
Jianfeng Feng, Sussex University (UK)
Antonio Fernández-Caballero, Universidad de Castilla-La Mancha (Spain)
Jose Manuel Ferrández, Univ. Politécnica de Cartagena (Spain)
Kunihiko Fukushima, Tokyo University of Technology (Japan)
Carlos G. Puntonet, Universidad de Granada (Spain)
Manuel Graña Romay, Universidad Pais Vasco (Spain)
Oscar Herreras, Hospital Ramón y Cajal (Spain)
Gonzalo Joya, Universidad de Málaga (Spain)
Elka Korutcheva, UNED (Spain)
Dario Maravall, Universidad Politécnica de Madrid (Spain)
José Mira, UNED (Spain)
Javier Molina Vilaplana, Universidad Politécnica de Cartagena (Spain)
Juan M. Moreno Aróstegui, Universidad Politécnica de Cataluña (Spain)
Luiza de Macedo Mourelle, State University of Rio de Janeiro (Brazil)
Nadia Nedjah, State University of Rio de Janeiro (Brazil)
Julio Ortega Lopera, Universidad de Granada (Spain)
Alberto Prieto Espinosa, Universidad de Granada (Spain)
John Rinzel, New York University (USA)
Eduardo Ros, Universidad de Granada (Spain)
Javier Ruiz-del-Solar, Universidad de Chile (Chile)
Moisés Salmerón Campos, Universidad de Granada (Spain)
Eduardo Sánchez Vila, Universidade de Santiago de Compostela (Spain)
Francisco Sandoval, Universidad de Málaga (Spain)
José Santos Reyes, Universidade da Coruña (Spain)

Luis Manuel Sarro Baro, UNED (Spain)
Jordi Solé-Casals, Universitat de Vic (Spain)
Emilio Soria Olivas, Universitat de València (Spain)
Ramiro Varela Arias, Universidad de Oviedo (Spain)
Marley Vellasco, Pontificia Universidade Catolica (Brazil)
Michel Verleysen, Université Catholique de Louvain (Belgium)
Changjiu Zhou, Singapore Polytechnic (Singapore)

Table of Contents, Part II

Nets Design

Applications in Robotics

Sources Separation

Genetics Algorithms

Soft-Computing

Images

Medical Applications

Other Applications

Table of Contents, Part I

Biological Models

Functional Models

Learning

Self-Organizing Systems

Artificial Intelligence and Cognition

Bioinspired Developments

FPGA Implementation of a Perceptron-Like Neural Network for Embedded Applications

Eva M. Ortigosa, Antonio Cañas, Eduardo Ros, and Richard R. Carrillo

Dept. of Computer Architecture and Technology,
ETS Ingeniera Informática. University of Granada, E-18071 Granada, Spain
{eva,acanas,eros,rcarrillo}@atc.ugr.es

Abstract. In this work we present several hardware implementations of a standard multi-layer perceptron and a modified version called extended multilayer perceptron. The implementations have been developed and tested onto a FPGA prototyping board. The designs have been defined using a high level hardware description language, which enables the study of different implementation versions with diverse parallelism levels. The test bed application addressed is speech recognition. The contribution presented in this paper can be seen as a low cost portable system, which can be easily modified. We include a short study of the implementation costs (silicon area), speed and required computational resources.

1 Introduction

The work presented in this paper addresses the study of the implementation viability and efficiency of Artificial Neural Networks (ANNs) onto reconfigurable hardware (FPGA) for embedded systems, such as portable real-time speech recognition systems.

ANNs are widely used for diverse applications, such as classification problems, because of their capability of learning from the input patterns, in order to use the acquired knowledge for new input patterns. There are different, well known ANN models, which exhibit similar performance levels in classification problems [1]. We have focused on the Multi-Layer Perceptron (MLP) [2], and we propose the implementation of a modified version called eXtended Multi-Layer Perceptron (XMLP) motivated, on one hand, by its hardware implementability (silicon area and processing speed), and on the other hand, by speech recognition characteristics that usually make use of two dimensional processing schemes [3].

An interesting feature of the ANN models is their intrinsic parallel processing strategies, although in most of the cases, the final implementation of the ANN is done through sequential algorithms (that run on single processor architecture) and do not take advantage of this intrinsic parallelism. In this paper we introduce two implementation versions, a parallel and a sequential design of a standard MLP and XMLP. This illustrates how easy is to tackle the silicon area vs. speed trade-off, when the design is defined with a high level Hardware Description Language (HDL).

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 1-8, 2003.

Learning is carried out off-line, but using a software version of the ANN that take into account the bit-depth of the different variables and the computation precision of the different stages. Therefore, the final classification results are exactly the same as the ones obtained from the hardware version.

2 Perceptron-like Neural Network

2.1 Multi-Layer Perceptron (MLP)

The MLP is an ANN with processing elements or neurons organized in a structure with several layers (Fig. 1): an input layer that is simply composed by the input vector, some hidden layers and an output layer (in which only one winning node is active for each input pattern for classification problems).

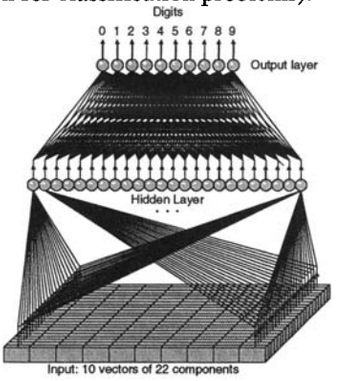


Fig. 1. Example of the MLP for isolated word recognition

Each layer is completely connected with its adjacent; there are no connections between non-adjacent layers and no recurrent connections. Each of these connections is defined by an associated weight. Each neuron computes the weighted sum of its inputs and applies an activation function (f(x) in Fig. 2) that forces the neuron output to be high or low. In this way, the MLP obtains an output vector from an input pattern, propagating forward the output of each layer. The synaptic weights are adjusted through a supervised training algorithm called backpropagation [2].

In the literature can be found different activation functions used in the MLP to transform the activity level (weighted sum of its inputs) to an output signal. The most frequently used is the sigmoid, although there are other choices such as a ramp function, a hyperbolic tangent, etc. All of them are continuous functions with a smooth S wave form that transform an arbitrary large real value (positive or negative) to another value in a much shorter range.

We have used the sigmoid activation function. The generic expression depends on three parameters f'_0, f_{max} y f_{min}: f'_0 is the slope of the function in $x = 0$, f_{max} is the

maximum value and f_{min} is the minimum. We have chosen the standard configuration ($f'_0 = 1/4$, $f_{max} = 1$ and $f_{min} = 0$). Fig. 2 represents the sigmoid function and the ramp function with parameters $f'_0 = 0.75$, $f_{max} = 1.5$ y $f_{min} = -0.5$.

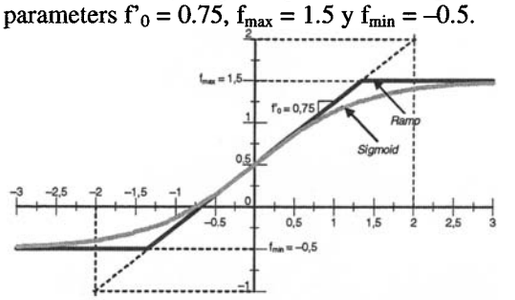


Fig. 2. Examples of activation functions: sigmoid and ramp functions

2.2 Extended Multi-Layer Perceptron (XMLP)

The XMLP is a feed-forward ANN with an input layer, up to two hidden layers and an output layer. The difference with respect to the MLP is that each layer can be configured as a two dimensional one and we can define the connectivity patterns of the neurons to restricted neighborhoods. This characteristic is very interesting for the speech recognition applications because in most cases we need to take into account a temporal environment of input vectors (for instance in phoneme recognition) or even it may be necessary to process the vectors corresponding to the whole pronunciation (complete word recognition). In these cases, the x axis can be used to control time, while the y axis takes into account all the features extracted from a single sample.

The size of each layer and its partial connectivity are defined by six parameters in the following form:

$$x(g_x, s_x) \times y(g_y, s_y). \tag{1}$$

x e y indicate the sizes of the axes, g and s define a group of nodes and a jump (step) between two consecutive groups in the X axis (g_x, s_x) and in the Y axis (g_y, s_y). To a node with index i in the X axis of layer k, only are connected neurons of layer k-1 of the i group of X axis, except if $g_x = x$ in layer k, in which case all the neurons of layer k are connected to all the neurons of X axis in layer k-1. The same is used to define the connectivity pattern in the Y axis. In this way, we use weights in rectangular neighborhoods that can overlap between themselves. Besides, it must be taken into account that the jump (s) in a layer cannot be larger than the group size (g). That means simply that all the neurons should have any connections. This structure is shown in Fig. 3.

With this architecture it is possible to implement networks of the kind Scaly Multi-Layer Perceptron (SMLP) used in isolated word recognition [4], and even of the type *Time Delay Neural Network* (TDNN), used in phoneme recognition [5].

Therefore, the MLP can be considered a particular case of XMLP when $g_x = x$, $s_x = 0$, $g_y = y$ and $s_y = 0$, in all the layers.

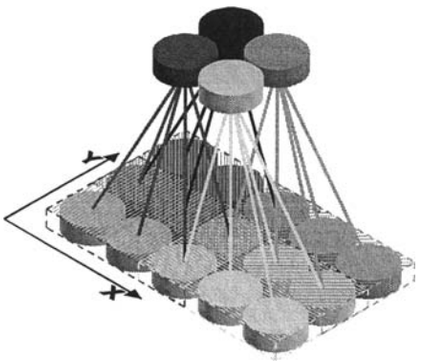


Fig. 3. Structure of a XMLP layer and its connections to the next layer. In this example, the lower layer, with 15 neurons, is defined by the parameters 5(3,2) × 3(2,1), i.e. 5 neurons in the X axis grouped in clusters of 3 with a jump of two neurons between groups, and 3 neurons in the Y axis grouped in clusters of 2 elements, with a jump of one neuron between groups. Because there are two groups in the X axis, in the following layer there must be 2 neurons in this axis, the same is valid for the Y axis

3 Hardware Implementation

To illustrate the hardware implementation of the MLP/XMLP system we have chosen a specific application related with speech recognition. Let us focus on *a voice controlled phone dial system* (this can be of interest for drivers that should keep their attention in driving). This particularizes the perceptron parameters to be implemented and the word set of interest (numbers from 0 to 9). Nevertheless, there are many other applications that require embedded systems in portable devices (low cost, low power and reduced physical size) such as voice controlled tools or robots, vehicles equipments (GPS navigator interface), toys, handicap people aids, etc.

For our test bed application we need a MLP/XMLP with 220 data in the input layer (10 vectors of 22 parameters) and 10 output nodes in the output layer (corresponding to the 10 recognizable words). And after testing different architectures [6] the best results (96.83% of correct classification rate) have been obtained with 24 nodes in the hidden layer and the connectivity of the XMLP defined by: 10(4,2)×22 in the input layer and 4×6 in the hidden layer.

For the MLP/XMLP implementation we have chosen a two's complement representation and different bit depths for the stored data (inputs, weights, activation function, outputs, etc). In the next section we introduce the discretization details.

3.1 Discretization

We have to limit the ranges of different variables:

- Inputs to the MLP/XMLP and the output of the activation function. Both must have the same range to easily manage multiple layers processing. (We have chosen 8 bits).
- Weights (8 bits).

Input to the activation function. The activation function is defined by a Look-Up-Table (LUT), but we have to specify how to store the useful values. In order to choose the bit precision of these values we have to take into account the maximum value reachable with the weighted sum of a neuron inputs. As the maximum number of entries to a neuron is 220, the maximum absolute value reachable with the weighted sum of a neuron inputs will be 3548380 (result of 127(max. input) × 127(max. weight) × 220(entries)). Therefore we need 23 bits for the input of the activation function in order to represent all the values between -3548380 and 3548380. But really, we do not need to store 2^{23} values. We could sub-sample it, but even with 14 bits we would need to store 2^{14} values. Nevertheless, if we take into account the sigmoid waveform we can store the significant values of the function with a much smaller LUT. In the sigmoid function (Fig. 2), most of the values are repeated, only the transition zone needs to be stored. Concretely, with 8 bits depth only 165 values of the LUT have values different of 0 or 127. Therefore, we could only store these 165 values with a small LUT addressable with just 8 bits. This can be done by implementing a specific address decoder.

After taking all these discretization simplifications the model achieves similar classification results. For instance, in phoneme recognition application with the MLP we obtained 69.33% of correct classification with the continuous model and 69.00% when using the discretized model.

3.2 Field Programmable Gate Arrays (FPGAs)

FPGA devices are composed by large arrays of programmable logic gates, and other embedded specific purpose blocks, such as memory blocks, multipliers, etc.

The internal structure of a typical FPGA consists of a regular matrix of configurable logic blocks (CLBs), surrounded by programmable input/outputs blocks (IOBs). The CLBs are interconnected by intermediate routing switches. Embedded configurable on-chip RAM memory blocks are also available on modern FPGAs, which can be used to store LUTs (as the activation function of the MLP/XMLP nodes or the synaptic weights).

FPGA can be configured as custom co-processors for a particular task. In this way we can design specific hardware for any application enabling the exploitation of the inherent parallelism of the implemented algorithms.

The hardware versions of MLP and XMLP have been implemented and tested on a Celoxica [7] RC1000-PP PCI prototyping board containing a single Xilinx [8] Virtex 2000E FPGA, and 4 RAM memory banks. The memory is accessible by the FPGA and by the PCI bus. It can be used to exchange data between the computer and the co-processor. In this FPGA a CLB element consists on 4 similar slices, with fast local feedback within the CLB.

Two versions of the MLP/XMLP have been described using Handel-C [7]: a high level hardware description language, which is a synthesis language based on the

ISO/ANSI-C. The development tool DK1 enables the easy description of algorithmic based systems, that are more difficult to address with other HDLs, such as VHDL or Verilog.

3.3 Implementation Characteristics

In this section we present the implementation characteristics of sequential and parallel versions of the MLP and the XMLP networks. Simplified block diagrams of the two versions, illustrating the basic differences between the sequential and the parallel versions, are shown in Figs. 4.a and 4.b. The parallel version computes the 24 hidden nodes in parallel, dedicating specific circuitry to each of them. Only the activation function is consulted sequentially in a single LUT.

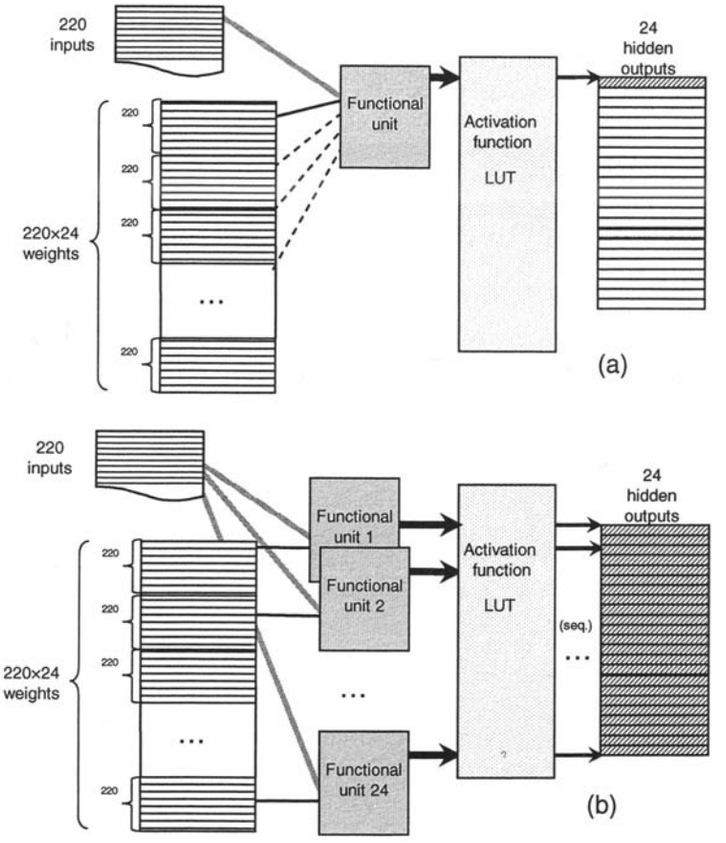


Fig. 4. Block diagrams of the sequential version (a) and the parallel version (b) of the MLP. Computations of the hidden layer (24 neurons) require 220x220x24 multiplications to be done sequentially with a single functional unit (a). Computations of the hidden layer are done in parallel with 24 functional units (b)

The systems have been designed using Handel-C with the development environment DK1.1 to extract an EDIF file. This has been finally compiled using the synthesis tool *Xilinx Foundation 3.5i* [8].

In Table 1 we present the implementation characteristics obtained after synthesizing the sequential and parallel versions of the MLP. We indicate the following parameters: number of gates, number of slices (Xilinx devices), the occupation percentage of a Virtex-E 2000 FPGA (that we have used to test the final implementations), the RAM size in bits, the maximum clock period, the number of clock cycles required for each input vector (220 input components) evaluation, an the total time consumed for each input vector evaluation.

Table 1. Implementation characteristics of the sequential and parallel designs of the MLP

Design MLP	# gates	# slices	% occup.	RAM (bits)	Clock (ns)	# cycles	T. comp. (μs)
Sequential	136731	2756	14	49648	63.615	5634	358.406
Parallel	208824	6343	33	48160	60.734	294	17.856

It can be seen that the sequential version requires less occupation than the parallel one in which specific circuits have been implemented for each hidden node. In the sequential version all the computations are carried out by the same functional unit. In contrast, the parallel version requires the synthesis of 24 functional units to compute the hidden layer in parallel.

The computing time for each input vector is much shorter (20 times) in the parallel version mostly due to the reduced number of clock cycles needed for an input vector evaluation. In this way we are taking advantage of the inherent parallelism of the ANN computation scheme.

Table 2 presents the implementation characteristics of the sequential and parallel versions of the XMLP.

Table 2. Implementation characteristics of the sequential and parallel designs of the XMLP

Design XMLP	# gates	# slices	% occup.	RAM (bits)	Clock (ns)	# cycles	T. comp. (μs)
Sequential	102182	2164	11	33440	66.125	2511	166.040
Parallel	183953	6169	32	33440	76.656	164	12.571

In this case, we observe that also the occupation rate of the parallel version increases dramatically (from 11% in the sequential design to 32% in the parallel design). And again the computing time of the parallel version is much shorter (13 times).

Finally, comparing Tables 1 and 2 we see how XMLP requires less resources than the MLP and achieves a faster computation (it reduces the computing time in 5 microseconds for the parallel version). The advantages exhibited by the XMLP are due to the partial connectivity patterns, which reduce the number of multiplications from 5380 in the case of fully connected configuration (MLP) to 2112 in the case of

XMLP with the configuration explained in Section 3. It is also observed the XMLP reduces the storage RAM requirements, again because it requires less connection weights to be stored.

4 Conclusions

We have presented the complete implementation of a perceptron-like neural network. We have compared the implementation of two ANN models, a standard MLP and a modified version of it with restricted connectivity patterns. We have designed the two models using a high level Hardware Description Language (Handel-C) in order to easily compare sequential and parallel designs of the two network models.

In order to optimize the implementation we have studied how to reduce the storage requirements for the activation function. The different designs characteristics show that is relatively simple to parallelize ANN, and reduces the computational time for each input vector dramatically.

The parallel designs have been defined through the Handel-C *"par"* directive that forces the implementation of dedicated circuits for a certain part of an algorithm to be computed in parallel. But it must be noted that since the designs have been defined using a high level Hardware Description Language the automatic synthesis pathway is long and optimizations in the DK1 compiler or the Xilinx synthesis tool may affect the design characterizations shown in Tables 1 and 2.

For the speech recognition application we obtain a correct classification rate of 96.83% with a computation time around 12.5 microseconds per sample, which fulfills by far the time restrictions imposed by the application. Finally, taking into account the size of the designs, all of them would fit into a low cost FPGA (less than 5 $). Therefore, the presented implementation can be seen as a low-cost design to be embedded in a portable speech processing platform (for voice controlled systems).

References

1. Maren, A., Harston, Pap, R.: Handbook of neural computing applications. Ac. Press, (1990)
2. Widrow, B., Lehr, M.: 30 years of adaptive neural networks: Perceptron, Madaline and Backpropagation. Proceedings of the IEEE, Vol. 78, No. 9, pp.1415-1442 (1990)
3. Cañas, A., Ortega, J., Fernández, F. J., Prieto, A., Pelayo, F. J.: An approach to isolated word recognition using multilayer perceptrons. LNCS 540, pp. 340-347 (1991)
4. Krause, A., and Hackbarth, H., Scaly Artificial Neural Networks for Speaker-Independent Recognition of Isolated Words, Proc. of the IEEE Int. Conf. On Acoustics, Speech and Signal Processing, ICASSP '89, pp.21-24 (1989)
5. Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., Lang, K.: Phoneme Recognition Using Time-Delay Neural Networks. IEEE Transactions on Acoustics, Speech, and Signal Processing. Vol. 37, No.3, (1989)
6. Cañas, A., Ortigosa, E. M., Díaz, A. F., Ortega, J.: XMLP: a Feed-Forward Neural Network with Two-Dimensional Layers and Partial Connectivity (submitted IWANN'2003)
7. Celoxica, http://www.celoxica.com/
8. Xilinx, http://www.xilinx.com/

NSP: A Neuro-symbolic Processor

Ernesto Burattini[1], Massimo De Gregorio[2], Victor M. G. Ferreira[3], and Felipe M. G. França[4]

[1] Dipartimento di Scienze Fisiche, Universitá degli Studi di Napoli "Federico II" Complesso universitario di Monte Sant'Angelo, Via Cinthia, I80126, Napoli, Italy
ernb@na.infn.it
[2] Istituto di Cibernetica "Eduardo Caianiello", CNR, via Campi Flegrei 34 Comprensorio Olivetti Ed. 70, I80078 Pozzuoli (NA), Italy
m.degregorio@cib.na.cnr.it
[3] Department of Informatics Graduate School of Information Science and Electrical Engineering, Kyushu Univ. 61 Kasuga Koen, Kasuga, 8168580, Fukuoka, Japan
goulart@c.csce.kyushu-u.ac.jp
[4] Programa de Engenharia de Sistemas e Computaçcãao, PESC/COPPE Universidade Federal do Rio de Janeiro, Caixa Postal 68511 21945970 Rio de Janeiro, RJ, Brazil felipe@cos.ufrj.br

Abstract. This paper presents an implementation methodology of weighted ANNs whose weights have already been computed. The validation of this technique is made through the synthesis of circuits implementing the behaviour of specialised ANNs compiled from sets of logical clauses describing different logical problems. A Neuro–Symbolic Language (NSL) and its compiler[5] have been designed and implemented in order to translate the neural representation of a given logical problem into the corresponding VHDL code, which in turn can set devices such as FPGA (Field Programmable Gate Array). The result of this operation leads to an electronic circuit called NSP (Neuro–Symbolic Processor) that effectively implements a massively parallel interpreter of logic programs.

1 Introduction

One of the main obstacles on implementing weighted binary ANNs in hardware is its impact in both silicon area (A) and time of propagation (T) – the AT^2 parameter. Nevertheless, weightless ANNs, or Boolean neural networks, can be naturally accommodated by existing digital systems, i.e., implementations in conventional hardware or software.

Digital implementations of weighted ANNs by use of the multiplication–summation–thresholding's scheme are very expensive in terms of silicon area and propagation time. Some previous works have focused on optimising parts of this computation, like the comparison mechanism [1], varying the depth (maximum number of layers from one input to one output) and size (number of neurons of the ANN) of the comparison neuron net, all seeking to minimise AT^2.

[5] Copyright n. 001109/D801353 - 23/12/98.

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 9-16, 2003.

This work focuses on reducing, as much as possible, the size of neurons (in terms of silicon area), what may also lead to delay reductions, by use of a simple methodology for binary digital implementations of ANNs. The method consists of substituting the traditional scheme of summation–evaluation of the threshold by a digital circuit mimicing the functionality of the whole neuron. The proposed synthesis mechanism can be naturally integrated into automatic or semi–automatic CAE tools and will be shown how this strategy may lead to great area and delay reductions [2].

In the following sections, we introduce both the neuron and the neural network models, then we show how monotonic and non-monotonic logical inferences can be implemented by means of ANNs and, finally, we describe the compiler that translates the whole network into the respective VHDL code.

2 The ANN model

The neuron model adopted here is the *Weighted–Sum non–Linear Thresholded Element* of McCulloch and Pitts [3]. Its state transition function is defined as follows:

$$p_i(t+1) = \mathrm{I}\left[\sum_{j=1}^{k}(a_{i,j}\cdot p_i(t) - s_i)\right];\ \mathrm{I}[x] = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases} \tag{1}$$

where $p_i(t)$ represents the state (1 or 0) of neuron $\mathbf{p}_i$ at time t, $a_{i,j}$ the coupling coefficient or weight between neurons $\mathbf{p}_i$ and $\mathbf{p}_j$, whose values may be positive (excitation) or negative (inhibition), s_i the threshold of neuron $\mathbf{p}_i$.

One may look for logical operations that may be performed by means of elements defined as in (1), where each element of this kind provides a localist neural representation of a propositional literal from some set $P = \{p_1, \ldots, p_n\}$. As stated in Aiello *et al.* [4], the two possible truth–values of a literal p are represented by means of two distinguished neurons $\mathbf{p}$ and $\neg\mathbf{p}$: the first is activated if and only if the corresponding literal is supposed to be true, the second if and only if it is supposed to be false. In the latter case of course we are entering the field of non–monotonic logic. The presence of both $\mathbf{p}$ and $\neg\mathbf{p}$ allows us also to check if an explicit contradiction arises on the basis of previous inferences. Inactivity of these neurons means that we do not have information about the truth–value of the corresponding literal. A similar approach has been proposed in the past by von Neumann [5] (*Double Line Trick*) to take into account non–monotonic behaviour [6].

The kind of production rule we shall consider here is a conditional expression of the form

$$p_1 \wedge \ldots \wedge p_k \rightarrow c \tag{2}$$

and can be represented as a net having k neurons $\mathbf{p}_1$, ..., $\mathbf{p}_k$ connected to a neuron $\mathbf{c}$ (see Fig. 1) with the following settings: $a_{i,c} = 1$ where $1 \leq i \leq k$ and the threshold of $\mathbf{c}$ $s_c = k - \epsilon$ where $0 < \epsilon < 1$.

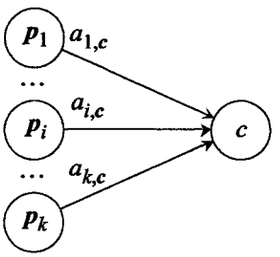


Fig. 1. Neural rule model

Using this representation of rules as basic building block, one can design a neural production system, organised into five different layers of neurons, capable of carrying out search on a knowledge base of facts and production rules. A specific example of such system, for a set of three rules, is represented in Fig. 2.

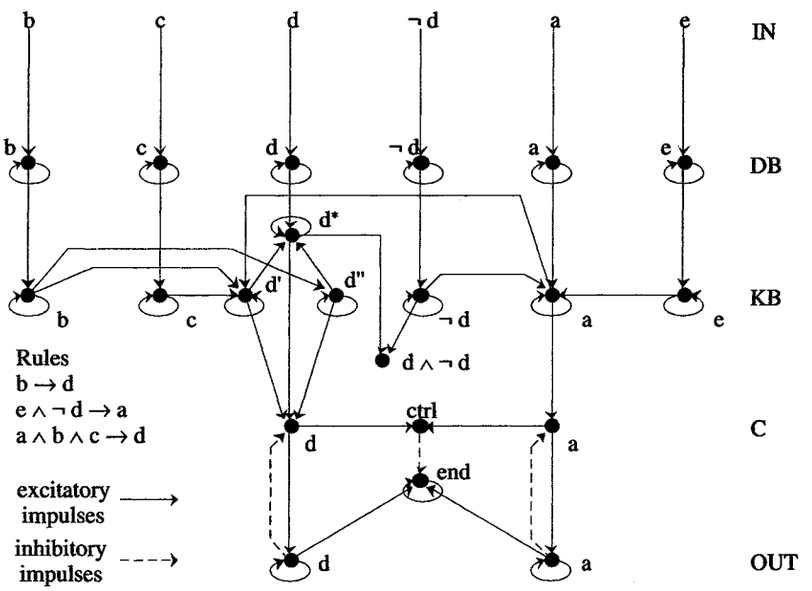


Fig. 2. NFC – Neural Forward Chaining

The neural network model reported in Fig. 2, is the one called NFC (*Neural Forward Chaining*) introduced in Aiello *et al.* [4] and it is based on a unified LCA (*Localist Connectionist Architecture*) approach [7] of rule–based systems proposed by Burattini *et al.* [8] in 1992.

The NFC model performs parallel forward inference processes for propositional rule–based systems. Moreover, for each set of logical clauses there exists the corresponding NFC neural network, which is generated by a compiler program based on the Neuro–Symbolic approach introduced by Aiello *et al.* [9]. The NFC computation captures the symbolic meaning of the corresponding set of logical clauses.

The non–monotonic NFC model (grounded on the same principles of monotonic one), is based on an appropriate use of inhibitions, to implement a parallel non–monotonic reasoning (in this case, inhibitions play a key role in asserting and retracting facts [10]).

The NSL (*Neuro–Symbolic Language*), partially introduced in [10] and reported in [11], allows us to express non–monotonic rules, such us "p_c is true if $P_\wedge$ is true, unless $Q_\vee$ is true" (where $P_\wedge$ is the conjunction $p_1 \wedge \ldots \wedge p_n$ of n literals and $Q_\vee$ the disjunction $q_1 \vee \ldots \vee q_m$ of m literals). This non–monotonic rule is denoted by the non–monotonic operator[6] $UNLESS(P_\wedge, Q_\vee, p_c)$ whose neural representation is reported in Fig. 3.

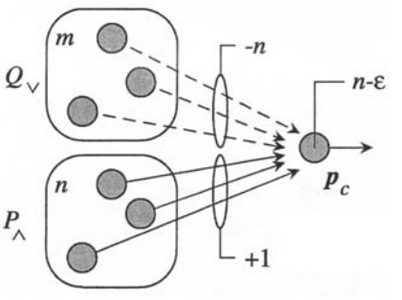


Fig. 3. $UNLESS(P_\wedge, Q_\vee, p_c)$ neural representation

3 A hardware implementation of a neural rule

The rules presented in the previous section, feed a compiler which, by its turn, builds a customised ANN architecture (neural rule) by defining the input weights of all interconnections and thresholds of each neuron defined in the process. From each set of input weights and threshold function of each neuron, an equivalent logic circuit is synthesised (see Fig. 4), based on each neuron's reactions to all possible inputs to it, up to the point in which the whole target ANN is converted into a set of interconnected digital circuit blocks (each block corresponding to one neuron) [2].

We used flip–flops because each neuron has to store its previous output, but the functionality of the neuron is enclosed in the Combinational Logic block.

[6] The set of NSL non–monotonic operators is reported in [11].

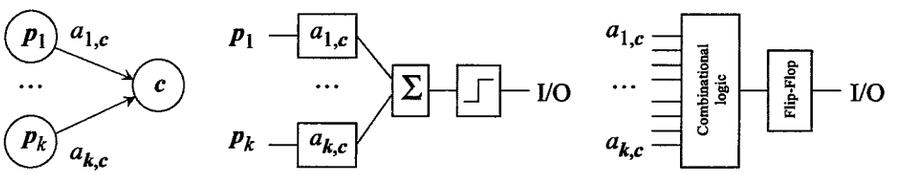


Fig. 4. The method proposes the substitution of the multipliers–adders–comparator's scheme by a combinational logic that implements a truth table

Differently from pRAM (*probabilistic RAM*) and DPLM (*Dynamically Programmable Logic Module*) models and implementations [12] [13], the neuron mapped in hardware may have no learning capabilities. It is assumed that the interconnection structure, all weights and neuron's thresholds are pre–defined and well determined by the compiler.

The resulting network is a digital circuit and its outputs are global composite Boolean functions of ANN's inputs. This combinational processing is fully parallel and potentially asynchronous, which may allow extremely fast execution.

Furthermore, the area used to synthesise the whole ANN is greatly reduced by a factor of 5.0 compared with actual implementations in hardware of the summators, multipliers and comparators (using a standard cell library).

4 From NFC to FPGA

Let be

$$\begin{aligned} b &\rightarrow d \\ e \wedge d &\rightarrow a \\ \neg d \wedge c &\rightarrow a \\ d \wedge a &\rightarrow b \end{aligned}$$

a set of rules expressing a logical problem. The rules can be written in terms of *IMPLY*s operators of the NSL in the following way: $IMPLY(b, d)$, $IMPLY(e \wedge d, a)$, $IMPLY(\neg d \wedge c, a)$, $IMPLY(d \wedge a, b)$.

The compiler, that has been designed and implemented by the authors of this paper, accepts a set of NSL operators. Each of them, is translated into an ANN that captures the symbolic meaning of the corresponding NSL operator. From the set of neurons forming the ANNs, the compiler produces the respective VHDL code. The correspondence between one neuron and the respective VHDL code does not depend on the particular neuron. The ANN is formed by homogeneous neurons whose architectures do not depend on the particular layer they belong to. For instance, the following code is produced for the neuron $\mathbf{e}_{KB}$ of the layer *KB*.

```
library IEEE;
use IEEE.std_logic_1164.all;
entity neuron_ekb is
```

```
port( clk, reset, edb : in std_logic; ekb : inout std_logic);
end
neuron_ekb; architecture SYN_USE_DEFA_ARCH_NAME of neuron_ekb is
... ... ...
... ... ...
end SYN_USE_DEFA_ARCH_NAME;
```

Once all the operators have been translated into the corresponding ANNs, the compiler generates (see below) the whole VHDL project of the NFC associated to the starting set of rules.[7]

```
library IEEE;
use IEEE.std_logic_1164.all;
ENTITY abt IS
    PORT( clk, reset, ain, bin, cin, din, ein, not_din : IN STD_LOGIC;
          aout, bout, dout , ctrl, S_end : OUT STD_LOGIC);
END abt;
ARCHITECTURE structural OF abt IS
component neuron_adb
    port( clk, reset, ain : in std_logic; adb : inout std_logic);
end component;
... ... ...
SIGNAL adb, bdb, cdb, ddb, edb, not_ddb : STD_LOGIC;
SIGNAL ckb, ekb, not_dkb, dkb, bkb, akb2, akb1, akbstar : STD_LOGIC;
... ... ...
BEGIN
U1: neuron_adb PORT MAP( clk, reset, ain, adb);
U2: neuron_bdb PORT MAP( clk, reset, bin, bdb);
... ... ...
aout <= SIG_aout;
... ... ...
END structural;
```

The derived NFC network can be both simulated or implemented on an FPGA device.

5 An example: a traffic light control problem

A prototype NSP was designed to solve a traffic light control problem [14]: on the basis of information provided by traffic sensors, concerning the number of cars approaching a junction from its branches, one has to decide which branches are to be assigned the green light and for how long.

The knowledge base, extracted from traffic control experts, was represented in terms of first–order production rules. Thus, the design of the NSP requires first transforming this set of rules into an equivalent set of propositional rules. It turns out that this can be done without having to introduce too many rules – where the criterion to judge whether the newly introduced propositional rules are "too many" is provided by the current possibilities of FPGA technology.

The NFC representation makes a significant difference from a computational point of view: in general, computation time is drastically reduced, since many rules are simultaneously evaluated, and only few computation steps are required.

The performances of the NFC for the traffic light control has been compared to the performance of the actual control system (based on the selection of prefixed plans) of a traffic junction of the city of Naples. Our system worked on

[7] For copyright reasons part of the code is missing.

the traffic flow data of an entire day. The following results were obtained: from the computational point of view, on the average, in order to evaluate all 220 rules (i.e., in order to obtain a new traffic light state on the basis of information collected by traffic sensors), 10 to 15 steps of computation are needed; from the traffic light control point of view, the average waiting time due to NFC control is reduced by a factor of 0.53 compared to the fixed time control (see Fig. 5). Furthermore, an FPGA hardware structure capable of supporting the NFC for controlling the junction is very inexpensive.

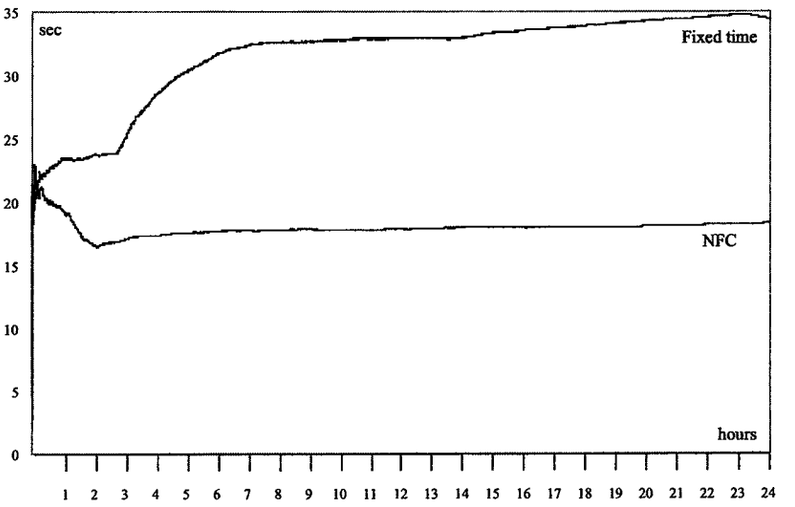


Fig. 5. Average waiting time – NFC control *vs* Fixed time control

6 Concluding remarks

The idea of designing computers using threshold elements or neurons is as old as the McCulloch and Pitts seminal paper [3]. Minsky in 1956 [15] and later in 1967 [6] reported on the possibility of implementing most operations performed by a von Neumann machine using McCulloch and Pitts neurons. Minsky [6] claimed that:

> *As the control over fabrication methods improves, we can expect the more delicate "threshold–logic" kind of circuit to play a large role.*

Minsky's expectations about technological improvements are partially met by the technology now available on the devices market: the FPGA. These processors enable one to implement logical structures like those reported in the present paper.

In conclusion, we may claim that NSP is a massively parallel interpreter of logic programs.

References

1. V. Beiu, "New VLSI complexity results for threshold gate comparison," *Proc. of SBRN'96* (1996).
2. V. M. G. Ferreira and F. M. G. França, "Weightless implementations of weighted neural networks," *Proc. of the IV Brazilian Symposium on Neural Networks – SBRN'97*, Goiânia, GO, Brazil, vol. II, 53–54 (1997).
3. W. S. McCulloch and W. Pitts, "A Logical Calculus of the Ideas Immanent in the Nervous Activity," *Bull. Math. Bioph.* 1943.
4. A. Aiello, E. Burattini, G. Tamburrini, "Purely Neural, Rule–Based Diagnostic Systems. Part I: Production Rules," *International Journal of Intelligent Systems*, **10**, 735-749 (1995).
5. J. von Neumann, "Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components," *Automata Studies*, Princeton (1956).
6. M. L. Minsky, "Computation: Finite and Infinite Machines," Prentice Hall (1967).
7. M. Hilario, "An Overview of Strategies for Neuro–Symbolic Integration," *in Sun R. and Alexandre F. (eds.)*, 1-6 (1995).
8. E. Burattini and G. Tamburrini, "A Pseudo–Neural System for Hypothesis Selection," *International Journal of Intelligent Systems*, 193-209 (1992).
9. A. Aiello, E. Burattini, G. Tamburrini, "Neural Networks and Rule–Based Systems," *Fuzzy Logic and Expert Systems Applications*, Accademic Press (1998).
10. E. Burattini, M. De Gregorio, G. Tamburrini, "Neuro–Symbolic Processing: Non–Monotonic Operators and their FPGA Implementation," *Proc. Sixth Brazilian Symposium on Neural Networks – SBRN 2000*, IEEE Press, 2–6 (2000).
11. E. Burattini, A. de Francesco and M. De Gregorio, "NSL: A Neuro-Symbolic Language for Monotonic and Non-Monotonic Logical Inferences," *Proc. Seventh Brazilian Symposium on Neural Networks – SBRN 2002*, IEEE Press, in press (2002).
12. C. K. Ng and T. G. Clarkson, "The VLSI design of a neural network processing element: the pRAM module," *Weightless Neural Network Workshop '93*, Computing with Logical Neurons, 134–138, (1993).
13. J. J. Vidal, "Implementing neural nets with Programmable Logic," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, **36**, 7, July (1988).
14. E. Burattini, M. De Gregorio, G. Improta, "A new traffic light single junction control system implemented by a neural network," in *Networks in Transport Applications*, Eds. V. Himanen, P. Nijkamp, A. Reggiani, Ashgate Pub. Ltd, Aldeeshot, England, 193–209 (1998).
15. M. L. Minsky, "Some universal elements for finite automata," *Automata Studies*, Princeton (1956).

Reconfigurable Hardware Architecture for Compact and Efficient Stochastic Neuron

Nadia Nedjah and Luiza de Macedo Mourelle

Department of Systems Engineering and Computation, Faculty of Engineering,
State University of Rio de Janeiro,
Rio de Janeiro, Brazil
{nadia, ldmm}@eng.uerj.br http://www.eng.uerj.br /~ldmm

Abstract. In this paper, we propose reconfigurable, low-cost and readily available hardware architecture for an artificial neuron. This is used to build a feed-forward artificial neural network. For this purpose, we use field-programmable gate arrays i.e. *FPGAs*. However, as the state-of-the-art *FPGAs* still lack the gate density necessary to the implementation of large neural networks of thousands of neurons, we use a stochastic process to implement the computation performed by a neuron. The multiplication an addition of stochastic values is simply implemented by an ensemble of *XNOR* and *AND* gates respectively.

1 Introduction

Artificial neural networks i.e., *ANNs*, are now well known [7]. They consists of a pool of relatively simple processing units, usually called artificial neurons, which communicates with one another through a large set of weighted connections. There are two main network topologies, which are *feed-forward* topology [7], [8], where the data flows from input to output units is strictly forward and *recurrent* topology [7], [8], where feedback connections are allowed. Artificial neural networks offer an attractive model that allows one to solve hard problems from examples. However, the computational process behind this model is complex. It consists of massively parallel non-linear calculations. Software implementations of artificial neural networks are useful but hardware implementations takes advantage of the inherent parallelism of *ANNs* and so should answer faster.

Reconfigurable *FPGAs* provide a re-programmable hardware that allows one to implement *ANNs* very rapidly and at very low-cost. However, *FPGAs* lack the necessary circuit density as each artificial neuron of the network needs to perform a large number of multiplications and additions, which consume a lot of hardware area if implemented using standard digital techniques.

Stochastic computing principles are well detailed in [6]. The motivation behind the use of stochastic arithmetic is its simplicity. Designers are faced with hardware implementations that are very large due to large digital multipliers, adders, etc.. Stochastic arithmetic provides a way of performing complex computations with very simple hardware. Stochastic arithmetic provides a number of benefits over other computing techniques such as very low computation hardware area and fault tolerance.

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 17-24, 2003.

Stochastic computing presents some disadvantages, such as the variance inherent in estimating the value of a stochastic signal and the increased number of clock cycles required to accomplish a given computation. The potential of massive parallelism driven by the small circuit areas involved may alleviate some of these disadvantages. Of particular importance is that, unlike binary radix arithmetic, the signal format in stochastic arithmetic is robust in the presence of noise/single bit faults and the hardware complexity of the computational elements is generally low [2], [3], [6].

Previous work on hardware implementation of stochastic artificial neural networks can be found in [1], [4]. In [1], the activation function is implemented using a lookup table. For small number of inputs, this can be done using a single lookup table. However, for more than 5 inputs, this implementation is impractical. So, for large number of inputs, the activation function must be partitioned in several 5-input functions. In contrast with this, our implementation implements the neuron activation function very efficiently and very accurately using a state machine, which size does not depend on the number of net input signals. In [4], the neuron architecture is sequential. That is the *input×weight* computation is one at a time. Unlike this, our neuron is fully parallel: all the products are computed at once in a single clock cycle.

This paper is organized as follows: in Section 2, we introduce stochastic computing principles and arithmetic; then in Section 3, we show how to generate an accurate bit-stream for digital inputs and vice-versa; subsequently in Section 4, we describe the proposed hardware architecture for a neuron; in Section 5, we present the overall architecture of the neural net as well as the outcome of such hardware. Finally, in Section 6, we discuss some time and area requirements of our implementation.

2 Stochastic signals and arithmetic

In a stochastic model, a number is represented by a long probabilistic bit-stream whose density of bits set to 1 is proportional to its numeric value. Stochastic computing uses simple functions to perform complex functions such as *multiplication*, *addition*, *subtraction*, *thresholding* and *activation*. (For more details about stochastic computing systems see [6].) There are mainly two stochastic representations: *(i) unipolar* wherein a number $A \in [0, 1]$ is mapped to a random binary variable with generating probability of A. So, number 0 is converted to bit-stream 00...0 and number 1 to bit-stream 11...1. *(ii) bipolar* wherein a number $A \in [-1, 1]$ is mapped to a random binary variable with generating probability of $A/2+½$. So, number -1 is converted to bit-stream 00...0 and number 1 to bit-stream 11...1. The two stochastic representations have quite different features and can be more suitable to one or another of the neural network topologies [7]. In the unipolar format, the information carried in a stochastic bit-stream B corresponds to $P(B = 1) = P_B$ whereas, in the bipolar format, B is represented by $2P(B = 1) - 1 = 2P_B - 1$. In the rest of this section, we deal with basic signal processing components required in artificial neural network hardware: stochastic multiplier, stochastic adder and stochastic subtracter. The architecture of the basic components for stochastic arithmetic is given in Figure 1.

Stochastic multiplier. Among the arithmetic operation, multiplication of two bit-streams is the simplest. It is performed by a AND gate and a XNOR gate in the case

of unipolar and bipolar representation respectively. In the unipolar case, this obvious as $P_C = P_A P_B$ and so $C = AB$. For bipolar signals, using a XNOR gate it can be shown:

$$P_C = 1-(P_A(1-P_B)+P_B(1-P_A))$$
$$= 1-\left(\frac{1}{2}-\frac{AB}{2}\right)=\frac{1}{2}+\frac{AB}{2}$$

and as $C = 2P_C - 1$, then we have $C = AB$.

Stochastic adder. The sum S of two or more bit-streams representing values in [0, 1] or [–1, 1] does not necessarily belong to [0, 1] or [–1, 1] for unipolar or bipolar signals respectively. So, it is not possible to perform addition of bit-streams exactly, independent of a scaling operation. A simple scaled addition consists of using a multiplexer, which randomly selects a given input I_i with some probability Sel_i such that $\Sigma_i\, Sel_i = 1$ will generate an output with a probability, which is a scaled sum of the input probabilities. Using a n:1 multiplexer, we have:

$$P_S = Sel_1 P_{I_1} + Sel_2 P_{I_2} + \ldots + Sel_n P_{I_n}$$

As for unipolar signals $P_{I_i} = I_i$, we then have $S = Sel_1 I_1 + Sel_2 I_2 + \ldots + Sel_n I_n$ which is a scaled sum of the inputs. For bipolar signals, we have:

$$S = 2P_S - 1 = 2\left(Sel_1\frac{I_1+1}{2} + Sel_2\frac{I_2+1}{2} + \ldots + Sel_n\frac{I_n+1}{2}\right)-1$$
$$= Sel_1 I_1 + Sel_2 I_2 + \ldots + Sel_n I_n + Sel_1 + Sel_2 + \ldots + Sel_n - 1$$
$$= Sel_1 I_1 + Sel_2 I_2 + \ldots + Sel_n I_n$$

A stochastic subtracter is simply a stochastic adder with the subtracted inputs negated as shown in Figure 1(c), wherein $D = I_1 + I_2 + \ldots + I_l - I_{l-1} - \ldots - I_n$.

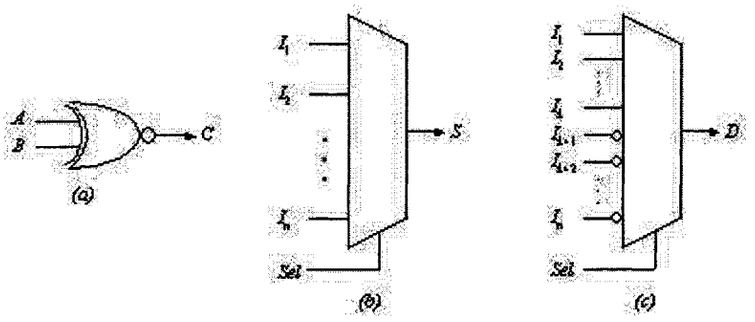


Fig. 1. Basic components for bipolar stochastic computing: *(a)* stochastic multiplier; *(b)* stochastic adder; *(c)* stochastic subtracter

3 Generating stochastic bit-streams from digital inputs

The purpose in this section is to introduce a *digital-to-stochastic converter*. It was first introduced in [5]. Other possible implementation can be found in [3], [4]. The converter generates a bit stream that represents a provided input value. To do so, the generator needs a statistically independent bit stream as input. Generally, this is yielded by a *pseudo-random bit sequence generators*, which we will describe later in this section. Now we concentrate on the architecture of the digital-to-stochastic converter.

The converter consists of sequence of pipelined *D-flip-flops*, each of which is preceded by a logic that computes either bit-wise AND or a bit-wise OR of the bit from the input value and the pseudorandom bit depending on whether the input bit is 1 or 0 respectively. The pipeline is supposed to yield a bit-stream in which the probability of a bit being set is equal to the input value. It has been showed [5], [1] that one can eliminate any inaccuracies due to random variance errors and so achieve a maximum accuracy with a digital-to-stochastic converter of length 2^{2p-2}, where p represents the number of bits in the binary input value.

The architecture of the converter is shown in Figure 2. In this figure, the *multi-taps linear feedback shift register* implements the pseudorandom sequence generator, which is explained next.

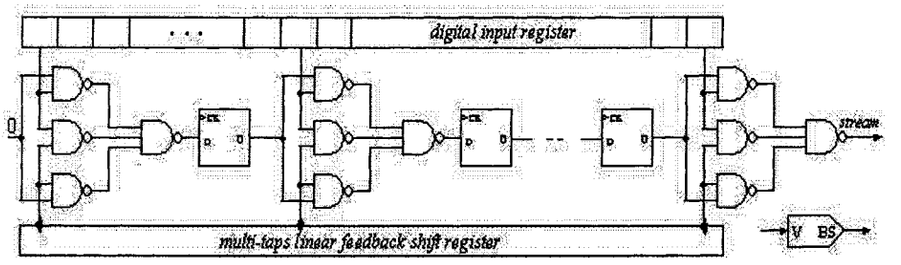


Fig. 2. Stochastic bit-stream generator and corresponding symbol

A fundamental element of a digital stochastic processing system is a source of pseudorandom noise. A source of pseudorandom digital noise consists of a *linear feedback shift register* or *LFSR*, described by first in [1] and by many others [2], *LFSRs* are very practical as they can easily be constructed using standard digital components.

Linear feedback shift registers can be implemented in two ways. The *Fibonacci* implementation consists of a simple shift register in which a binary-weighted modulo-2 sum of the taps is fed back to the input. Recall that modulo-2 sum of two one-bit binary numbers yields 0 if the two numbers are identical and 1 if not. The *Galois* implementation consists of a shift register, the content of which are modified at every step by a binary-weighted value of the output stage.

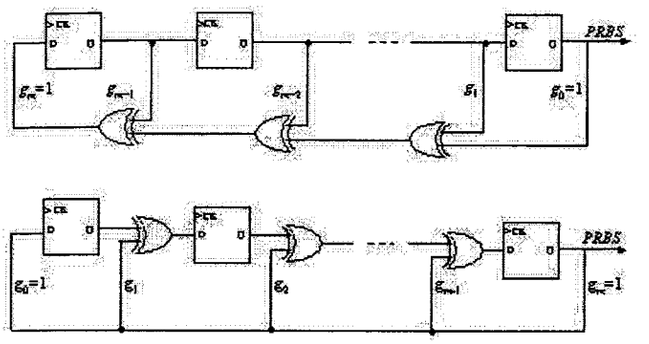


Fig. 3. Pseudorandom bit sequence generators – fibonacci vs. galois implementation

Left feedback shift register such as those of Figure 3 can be used to generate multiple pseudorandom bit sequences. However, the taps from which these sequences are yield as well as the length of the *LFSR* must be carefully chosen. (See [2], [6] for possible length/tap position choices).

A given bit-stream cannot directly be converted to a binary digital number, but one can generate an estimate for the probability. This is can be performed by counting the bits set in the stochastic stream. As probabilities can be negative, we use a up/down counter shown in Figure 4, wherein N, $abs(X)$ and $sgn(X)$ are the number of bits in bit-stream X, $X\langle N-2{:}0\rangle$ and the most significant bit of X.

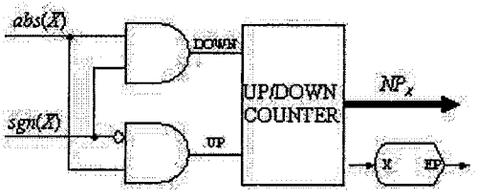


Fig. 4. Stochastic-to-probability estimator and its corresponding symbol

4 Neuron architecture

The computational process performed by a neuron is now agreed upon. A processing elements accepts N inputs i.e. $inputs_1$, ..., $input_N$ and yields a unique output, computed as follows:

$$output = \mathcal{A}\left(\sum_{i=1}^{N} weight_i \times input_i\right),$$

wherein $\mathcal{A}$ is an activation function that maps the weighted sum of the inputs to a neuronal output. In this paper, we use a continuous sigmoid-like activation function that is computed as follows, where K is a given threshold value:

$$\mathcal{A}(S,K)=\begin{cases}1 & \text{if } S>K\\ 0 & \text{otherwise}\end{cases}$$

The proposed architecture for the basic processing elements is shown in Figure 5. The inputs as well as the weights are converted to stochastic bit-streams. The generation of the weight and threshold bit-streams is performed inside the neuron, which allows one to have different weights and threshold values in distinct neurons. However, the bit-streams corresponding to the inputs are generated once for all, i.e. outside the processing elements.

The *weight×input* product is performed in serial manner using a single XNOR gate. The sum of the weighted sums is also obtained in a serial manner and implemented by a N:1 multiplexer. The random selection bits of the multiplexer is generated using a counter with a modulus equal to the number of inputs, driven to either increment or maintain its state each cycle based on a single random bit. The activation component is implemented using the state machine engineered by Bradley et Card [1]. Inputs and outputs of the component are stochastic bit-streams. The accuracy of the stochastic output depends on the total number of states.

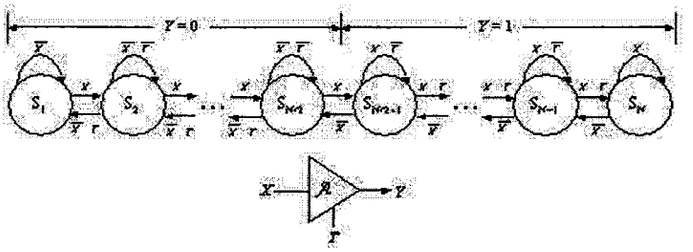


Fig. 5. State machine that implements sigmoid-like activation function and the used symbol

So a given neuron is constituted by N stochastic bit-stream generators, N XNOR gates, a stochastic N:1 multiplexer together with the corresponding selection logic, which a counter, an activation component and finally all the necessary routing to connect these components. The hardware architecture of the engineered neuron is shown in Figure 6. It is then clear that the hardware area required for the implementation of neuron depends on the total number of inputs N.

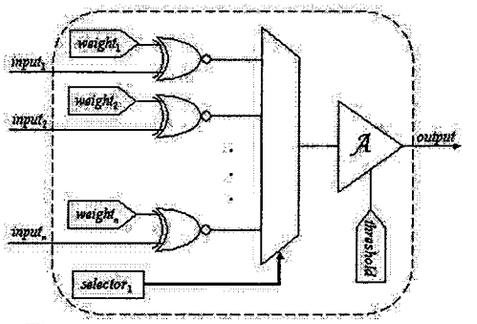


Fig. 6. Stochastic bipolar neuron architecture

5 Overall stochastic neural network architecture

In general, pattern of connections of PEs to form an artificial neural network can be classified as: *(i) feed-forward*, where the data flow from the input to output units is strictly feed-forward. The data processing extends over several layers of neurons, but no feedback is allowed; *(ii) recurrent*, where the data flow from the input to output neurons is a mixture of feed-forward and feedback connections. Recurrent topologies are used when dynamical properties of the network are of some importance and so need to be captured through the feedback connections.

In this paper, we concentrate on fully-connected feed-forward networks. To each input corresponds a stochastic bit-stream generator, which transforms the input real value in [−1, 1] to a bit-stream. The input is then fanned out to each neuron of the network first layer. On the other hand, for each neuron in the network last layer is associated a logic that estimates the probability of the output. The output probabilities behavior of the network hardware implementation is shown, in Figure 7, as function of the input probabilities and the threshold value. The state machine that implements the activation function has 32 states.

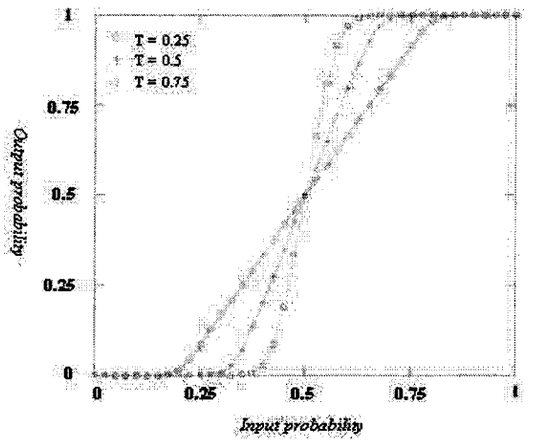


Fig. 7. Activation function behavior

6 Performance results

The entire design was done using the Xilinx Project Manager (version Build 6.00.09) [10] through the steps of the Xilinx design cycle. The design was elaborated using VHDL [9]. The *synthesis* step generates an optimised netlist that is the mapping of the gate-level design into the Xilinx format: *XNF*. The *programming* step consists of loading the generated bit-stream into the physical device. The design was programmed on device from VIRTEX-E family [10].

In Table 1, we relate the size of a single neuron for different numbers of inputs, the net size and the net delay. In the reported hardware implementations, the neural networks are squared fully-connected feed-forward nets of neurons. So the number of

the net layers is the same as the number of neuron input. Hence if N is the number of neuron inputs than the total number of neurons in the nets is simply N^2. In all of these implementations, the state machine for the activation function has 32 states.

Table1. Time and area requirements for different number of inputs

Number of input	Total number of neurons	Neuron size (CLBs)	Net size (CLBs)	Net delay (ns)
2	4	2	21	5.85
4	16	3	75	7.09
8	64	5	421	19.87
10	100	8	903	43.55
16	256	10	2560	87.45

7 Conclusion

In this paper, we presented reconfigurable compact hardware implementation for stochastic neural networks. So it exploits the inherent advantages of reconfigurable hardware, such as availability and low cost. The architecture is easily scalable. Very large neural nets, can be implemented across multiple FPGAs with very reduced extra effort. The processing within a single neuron is serial. However, the overall net architecture is inherently parallel, resulting in reduced signal delay propagation.

References

1. S.L. Bade and B.L. Hutchings, *FPGA-Based Stochastic Neural Networks – Implementation*, IEEE Workshop on FPGAs for Custom Computing Machines, Napa Ca, April 10-13, pp. 189-198, 1994.
2. B.D. Brown and H.C. Card, *Stochastic Neural Computation I: Computational Elements*, IEEE Transactions On Computers, vol. 50, no. 9, pp. 891-905, September 2001.
3. B.D. Brown and H.C. Card, *Stochastic Neural Computation II: Soft Competitive Learning*, IEEE Transactions On Computers, vol. 50, no. 9, pp. 906-920, September 2001.
4. M.V. Daalen, P. Jeavons, and J. Shawe-Taylor. *A stochastic neural architecture that exploits dynamically reconfigurable FPGAs*. Proceedings IEEE Workshop on FPGAs for Custom Computing Machines, pages 202--211, April 1993.
5. M.V. Daalen, P. Jeavons, and J. Shawe-Taylor. *A device for generating binary sequence for stochastic computing*, Electronics Letters, vol. 29, no. 1, pp. 80--81, January 1993.
6. B.R. Gaines, *Stochastic Computing Systems*, Advances in Information Systems Science, no. 2, pp. 37-172, 1969.
7. M.H. Hassoun, *Fundamentals of Artificial Neural Networks*, MIT Press, Cambridge, MA, 1995.
8. P. Moerland and E. Fiesler, *Neural Network Adaptation to Hardware Implementations*, In Fiesler E and Beale R eds., Handbook of Neural Computation, New York: Oxford, 1996.
9. Z. Navabi, *VHDL - Analysis and Modeling of Digital Systems*, McGraw Hill, Second Edition, 1998.
10. Xilinx, Inc. *Foundation Series Software*, http://www.xilinx.com.

e de y e f
e y e

s s s z
s 2 d s l z d d2

D o c ron ca cno og a o a ora ro c o n r a
o cn ca ar ag na c/ Dr ng /n
0 0 ar ag na S a n
{ i e e e R Rui u t e
www et u t e
ra n o r In r In gr r Sc a g n o an
0 r ang n G r an
{ i ue ii g e
www ii g e

A st t a r r n an ana og S n a on o a
n ra n wor a on a na cor o n wor co r
r a r o c a o on ron In rn ron an R n aw
c w c or a c con ro o or or a ng o
n n a an ng N ron a o r cr a ff r n a
q a on w c ro w a na c r or anc n wor
o con ro o n a n an agon a r o ac a
or c or a an or c r c o or or S r or
ac n a ca on r c r a a an ag a a ow n
n n con ro o o on an ffn w c a a
or a ca on w r n r a oa co n a on a cr ca ac or

n uc i n

t l l d p s t s t p l ls s ll
t l p d ts t t ll d l t d d l t
t l l s ts S l s s t p st s st d t d
d p d t t l s l l t d t t s t
t d s l t l t w t t l t s
tl t t l d ls d l p d st t t l l
s w s p ls s t sp l d s ll l s
ss t d t pl pp t s l s w t s t t d
pl s ll d t s s t s t l
lls pl d s t w st t t p d l
lls t ll s d w t t sp l d t l t s t
sp d t t l t s s t d t t s
t ls s t lls w p d l t p d tt
d p d t t l tw st ff ss d p s t t t l s l

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 25-32, 2003.

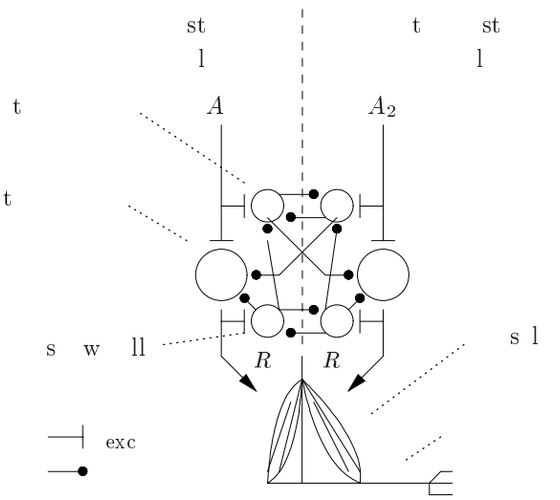
d ls d l t t d l p d t d st d l w
s d tt t ll d t st ds t z t
t d s d p p s s d p d t st ff ss d t
p s t t s d l st ts t s pl t l d
s l l t p s d p s l s d p t
s t pl t d l p s d ll d s d
t l l sp l d t s pl t s t d d t s d l
s sts p lls t l ps t s w st s l
t t d l w ld p s t p l t s w
d s t s s s l d ts t t st t w ld d p d
t d t t st t t t s t p s ts

u n w sc ip i n

t s s w s t d l s t t t s l
s t s s d t d l S pl t t s d s ts
t st s s pl t d pt l s p p s s s
w t d t s l tw d l s s d t l t
d t sp l d d t p s s s l t p s lls
st t d l s t s s w t s pl t t s d l w l d s
s s t t p s s lp t s t s d s w
lls t s d s w lls s ll s p s w t
t s d t l s t p d t d t p s t
t s st d t s tw s ll w
t t d ff t l t

— *c o o o*

w s t t t st t t *o* d *b o* t
p t s ls t sp t l s d s t st t *U* d
o sp t pp d l w l ts t t t st t
w s s pl tw p s d s s s
s d ff t s z d t s w t d ff t d t
t pp d l w t t st t l ts d ff t d $_2$ p s t
t p pp t t s d t t lls sp s l t
d t t t s w s t t tp t t s t
s l s t l d $_2$ t t s l t p
s ll lls w p d t t l s ls t w
t ts t t pp s t t d t pp s t t
ll s w lls d $_2$ d t t s t s
l d t l t t t t s l st s
t tw t s s w t d t
p ss w t t d ff t l t s st w d ls t
l tw t t s t s w l d d t l t l s
t t st t s t s pl t s w tw d



F S agra o

ts t t t s t p t s l d t t s
t dd t t pp s t t d t sp d s w
ll

$$\frac{}{} \quad A \quad {}_2 R$$

$$\frac{}{}^2 \quad {}_2 A_2 \quad {}_2 R_2$$

$$\frac{}{} \quad A \quad {}_2 R$$

$$\frac{}{}^2 \quad {}_2 A_2 \quad {}_2 R_2$$

$$\frac{R}{} \quad R \quad R \quad R_2 \qquad 6$$

$$\frac{R_2}{} \quad R_2 \quad {}_2 R_2 \quad R$$

i v syn sis

t s s t t s ts d l p d t d d s d t
t t l s s d s d p s s t w s s t s t s s
st d d ff t l t s s st t s l w
s t d pl t t t t lt d
s s t s pt t l s s t s p t s w d
t s t s s t d l lt pl s d t t s s ll t t
t lt d t p ts d t d p t

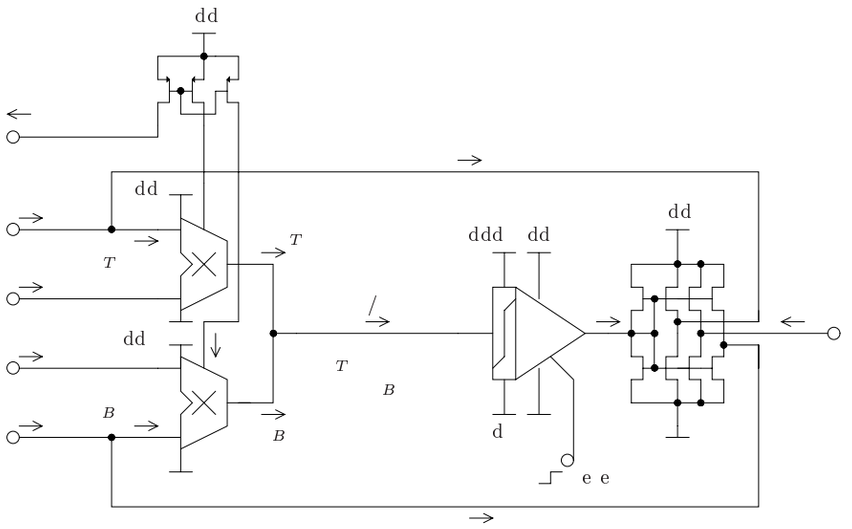
t d t d dd s d s st t s s s t pl t s
s pl d s t w t s pl t t w d t s
s t s l w w d t t s s l t t
d l t d t s d d s d t s t pl t
t s w ll d d d t s t t d w st l s d
t t p pl t t w w ll pl t s s t l
d d ff t l p t s p t l d s p ts d
tp ts s s s t t ll w s

p t p t ts t p s t d t t
p t tp t ts t t d t p s t

s s s t ss s p t t t t w l d s d w
w ll s d t l t t pl
s ws t s s l t t s p s d
l t t d tw lt pl s s d l s s st t s
s l p d t d s t l d s t t t
lt pl s d *A* s t t w w s p d d t t
t l t s t s t t d d s d t
w tw l d t l ts t d t s t lt pl s
t ws t d t p t d t s d s ts
s d t st l s t pp d l w l t t t t st t
d sp t l t t t d t p ts t t s
t tp t l t t st t p ts t lt pl
d w l t p ts t t s d lt pl d
tp ts t lt pl s dd d t t w t s d s p t
t l t t s t tp t t t t s t p t
st w t t t p t t t s t dd t
t lt pl s tp t w l ds

---

s w s t w t d l t d t
w pl ts t s t t t t t t
tp t p d s t ts l t w t s d t d
t s st t d sp t l d d t t d s t
tp t s l ss s d l t w w t p s t d
t ts t lt pl t p t w s d t p t
d s d t l s z d t w lt pl s
d t lt pl w s tp t s s l d t d p d t s
t d t s z l t d s s d t t t t w
s d l ss t t t 6 w s p p t s s ll
d l w p w s ppl w ts l t d l tw
ppl t s t l d s sw t s d t s t t t t
t t t t p l w p tt p s t s s
st t t d s s l t s w w ld



F 2 Sc a o ng n ron

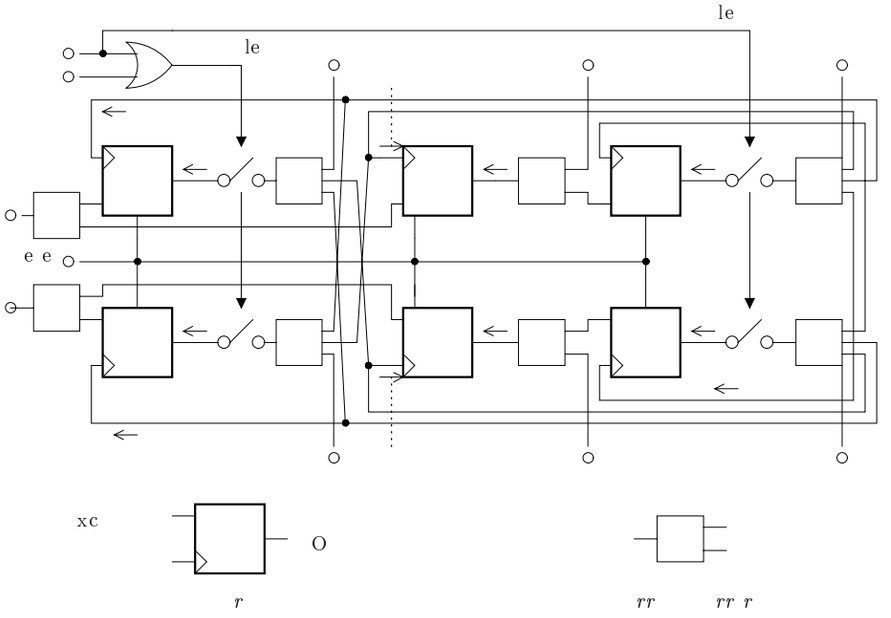
t w l tw pl t p p t t s t d s
t t s s t ll ws p l l s t *U* d
lo t t st t l s w s t s p t ts t t ts s
p l t ll ws t s s d s ll t tw s
ll d d l z d s t s tw l s

w v syn sis

pl t t t tw s d p ss
S l t s z s 6 d t p w d ss p t s
s lt lts t l d d t l p w s ppl
d s s d d t w l d s l d d t l p s
l p s l d p w s d tp ts t s s t s
p s s d t st l s t pp d l w l s st t s s t s
s s t l l tw t t l s st s st
t s t l p s l d s s t d t l p s s t s s d t
s s l l s t t tw w s p t s l
t ll st t t l p s s d t l d s l t s
d s w lls s t s d t st d t ff t t s
d d l lls t s st l d t ls t t t lls d p d
t l p l s
s ws t l s t sp l d s pl d d l t
w d t pl t d sw t s pl d t t tp t t
s d s w lls s d t l d s l t t p d

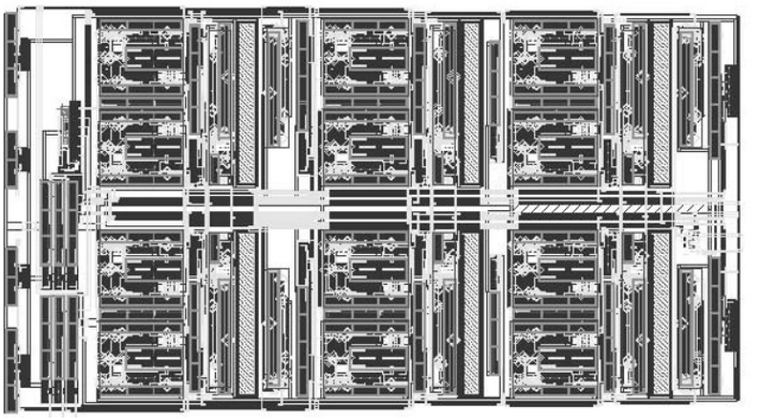
S S	N ron na
0 0	o on ron
0	o on ron In rn ron
	o on ron In rn ron R n aw
	on ro n

t s lls ts s w t l s l s t tw s
t t st t s s pl t s ts s d t st l s pp d l w
l ts t t st t s t s w t l d t l p w
s ppl s d s s d t t d sw t s t tw p s s
s t s s d t t p s t ts w d t
t d t t d l s ws t l tw l t



F S na cor n wor

st t t s d t t t s pl d
s t s w s d t l w l t
l t w l 2 2 d 2 t pp l s t t
lt pl s s pl d t l t t d s t pp p t w l t s
t s t t st t s pl d t l w p t

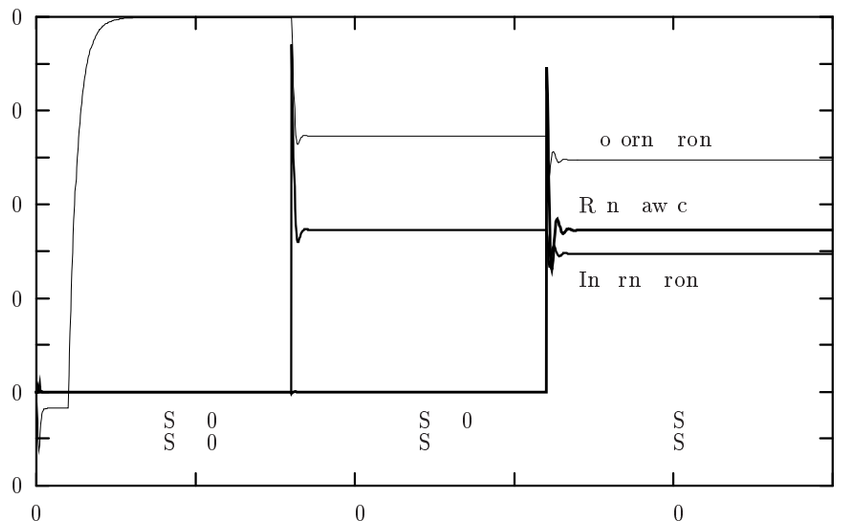


F N wor a o 0 x 0

d t ss t p p w t w s d s
s l t s t p d l s lts t t s l t t s
t w t l s t d t t t d tl st s l t S l t s
t tw d l d t p p s ws
t l t t s t s d s w lls
t t t p ts d $_2$ t t s p s t t
p s t d t t s l t s s t d l d l s l t
d ff t l s S d S_2 stl l t s t d l t
w l t s d ll s w lls ts d s s
w s t s pp l s w l t l d d
w ll s s d s w s d ff s t
s ttl t w s st t l t t s s d t
l t l t t s w s s t t st l
l st t d l s l t t t t t t d tl st w
l s t d st l p t p s l s t
d dd t d s st t s d t ffs t d w s

C nc usi ns

ppl t s w t s tw s d ll tw d t s
st t d l p t t s p l t s d t
d st d d p d p pl d s d t d l p t l
t ll s d st l ts p d t w t pl p
t s l st s sp l t t s d d l s p ll s S
w t s l t s l s t t l ss l t t s
t ld l l sp d s st s t d s p t t s
d w t t s w s l t t



F 5 N ron a con ro n

AC OW M

s w s d t s t d p d d U s d d l t
d t t t t U d
st t t t t S lt w t w s s t s z d

f nc s

G on a J x oo o ca o og Sa n r
o an oron o
o ng wang an S o n ar o n or ro o c con
ro In I on ro S
oc D on r ra J Gro rg S In r a oa co n a on a o
na c rc r ng ng o n o n In c R or S/ NS 00
r a ro S In rna ona G 0
anno I a ang o r a ran S rr n o r
In n n o D c ara r In I ran ac on on rc an S
II o 000
S S Sanc S n nc o ow o ag n gra or or g r c nc
S r ng c rr n o c n q In I ran ac on on rc
an S II o

e e ed d e e de f e e f GA d pp

t z l d d z

D o c ron ca cno og a o a ora ro c o
n r a o cn ca ar ag na a ra a ar / Dr ng /n
0 0 ar ag na S a n

e i g ut ie ti e u t e

A st t n w a roac o ar N ra N wor cr o
ro o a roac oc on NN n a on on r
con g ra ar war arc c r an DS cro roc or NN ar
ana ro r c o S or g ng r o an a r
na o o o o n n ra r a a a D na c q a
on an r o on a ana an r a n a on
arc c r ar cr n a r a o r an o n n
o n o o r o a n r o a n ro ff r n
a on nc n an nc ona o o

n uc i n

ll l l tw s t d d t s s t
s t ll l l tw s d ll l l t
w s U s l U t st p t t t s
p d s l l t t ll s t d l t lls
d d t d d s w t s st t d s t d
l l p ss w s p ss l t s t p p t d l t
t ll l t tw lls d t l l
l pl t l t p w t s ll
ps pl l l s st t t
sp d l ll t d lls l d t st d d d l p S pl
t t s s d t U p sl t d p t s w t l
d d t l l l ts p l s l d l t t pl t
t l ss t s s t s d p s ts
s ppl t s s d l d t d s l S d s
t s st t s t t d w d s t sts s t st
d t l s l p ss d ts pl t t t l sp p ss s
S t ls dw l d s s ff s w
p ss l t s s s d l s t t s d t s s
l l t d l p t d pl t t
w d s t d l s p s t d t s p p ll d l
l d s pt d t l l t t l t s st

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 33-40, 2003.

d d S st s t s l t d d
s l d t d ff p d t l l s d st d s pt t
st l t t d l p p s d t t t d s t l s
d l p t s t d t t pl t t dw
l d s d d t l s l p ss s S ff w
lt t t t s d s d t l t d l t t 6 ll
d ff t s l t s d s lts p s t d ll st t t d t s t
p p s d t t d t t d l s d ts s t s s
tl t s p p s s ll ws S t t d l
s d s d d l t d s l t sp d t s t d l s
st l t s st d d d l s d S t t t pl t t
d t s lts p s t d S t d t l s s S t t
s w t

C M in ys y

d s d ll l l tw s
t st t t d t t t t *f*

$$\frac{\qquad}{\overline{R}} \qquad A$$

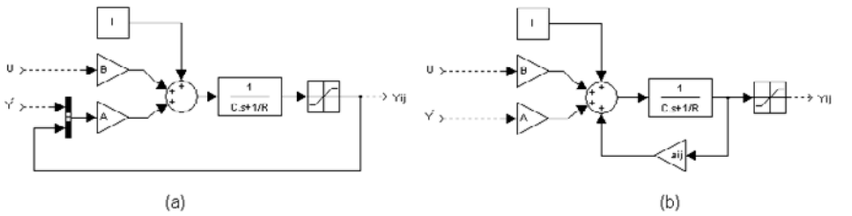
, ,

f –

w d t p t s p t tp t d st t l
ll sp t l d d st ll s
t w d d t t p s t t ll t tw d d
t p s t t ll l t t t ll s d t s t
st t w ts t pl t p ts d d *A* s t sp d t
pl t t tp ts lls ll l t t t
tp t sp ds t p w s l p t

$$\frac{\text{, } \qquad A \qquad \text{,}}{S \quad -}$$

t s p ss l t w t t st t t pl s w
t s t s t s t d s d t pl tl ss t st t
l t l tp t l t l ss t s l t
p l t t s t w t s pl d t
t l d l s s sts t l d l p
t ll l s s w t lt t d l
p p s d t s p p w s l t p ss s t d ff t l
t s s w d p ss d t d d ff t
d d t w d l



F a a ang or g na o ro o o ca on o or g na o

— $\overline{R}$ A ^{2}
, ,
; ,

— $\overline{R}$

— A ^{2}
, ,
; ,

i nc u i n s u i n n sy p ic s i i y

t t l s l t t d t t d s t s t s t s l t pl s t p t l t s l t d t s l t s t w s t t t t s t n w l t l s l t

∞

6

s t st d t s l t p t l t l t s l t p d d t l d t t t t st t s l t t tw t st t p ss d t t

γ γ ^{n}

^{n}
^{n} A
$^{\tau}$, ,
; ,

w

$\overline{}$ $\overline{R}$

d

γ $\overline{}$
$\overline{}$

w t d ff t s l t t s st st l t
st d d s t d ff t d t s d t s d s t d
s st s st l t t

e T d u o (d (d d
^{2} d o b bl f d ff olu o u o (

o
d o uou
fo

oof s d t d ff s l t t d d t ss pt
(

A
, ,
; ,

A
, ,
; ,

w s st t
s t w w t t t t
^{n}
γ γ ^{n} $^{\tau}$
$^{\tau}$
w w s t ss pt (w t

γ $\overline{}$

t s s lt w s t t ll s l t s l t d
t st l d p d tl t t l d t

e *T d u o (d (d d*
2 *d o b lob ll o ll bl f d ff olu o*
u o (

o
(o fo

oof s l t d ff s l t t d t ss pt *(*
w t t

γ γ A f

γ γ^{2} $^{\tau}$ A f A f
$^{\tau}$

pl w t t t w w t

γ γ^{2} $^{\tau}$
$^{\tau}$
A f A f $_{2}$ $_{2}$ A γ

d t l t s

^{n}
γ γ^{n} $^{\tau}$ ^{n} A f
$^{\tau}$ 6
^{n} 2
$^{\tau}A$ f $_{n}$ $_{\tau}$ $_{n}$ $_{\tau}$ A γ ^{n} $^{\tau}$
$^{\tau}$

w w s t ss pt *(* w w t
t 6 t t

γ A f A γ

$^{\tau}A$ f
$^{\tau}$

t t t sp t t s t l d t
t *A* t s st ts d γ s st t t
w t t t ll t s l t s t s st
s pt t ll st l d p d tl t t l d t
S l t st l s s t t d d t

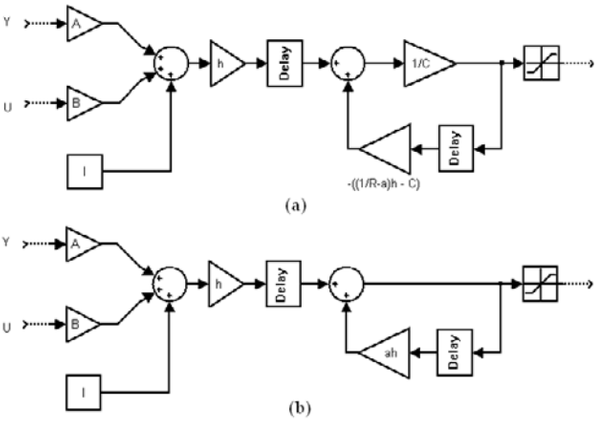
s st t s s pt t ll st l

$$\overline{R} \quad \overline{\quad} \quad \overline{R}$$

s pt t s l t d p ds l t l A
γ d
s pt t s l t s d t t t s
pl s t t t s st l t
t s st s st l t s l t w ll l t d t t f
ts l
t s st s st l t s l t d w st ps

A c i c u f i p n i n n si u i n su s

d l t s st s l d d pl t d s d ff t d t l d s d S p ss s l d t p p s d t t s t t l ss l t t t t s pl d ss t t l t R d s w t t s t d w t ll s s t z d t d p t s s

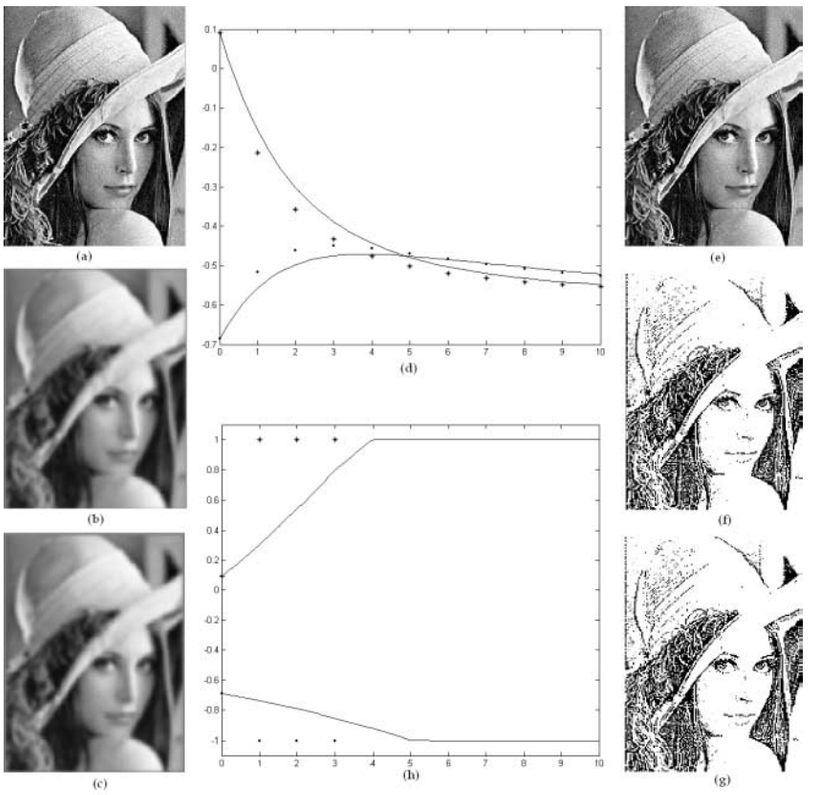


F 2 ro o c arc c r a o arc c r S ar c c r

ll t st t t t d t t s d l s t t st t d t s t t s t s st l t l t

$$\frac{R}{R}$$

w s t d s t z t st t d t s s st l t l t t l t d s t d l d t l t p p l w t t p p s d d l w s t l l st ss d t t st t t tw pl s d t l p ss d ff s d d d t t t w t l d t pl t s w d s pl s t t l t t tw l d ff t l s t st t d d t t s ss t s l t t d d t t p ss st l s st d d t f t ls s d



F a an or g na ag D ff on w 0 an 0 ra on c D ff on w 0 an 0 ra on wo x o on or ag an c g c on w 0 an 0 ra on g g c on w an ra on wo x o on or ag an g

C nc usi ns

t s p p w l t d d s t d l s p p s d d t ff lt t d ff t t t d t l d ls st p s t d l d l d s d d ff t t t s l t t d ff t d st d t st l t d s pt t st l t t s st ff d t st t t d pp t t d l l s ts l t t t z t t ll l l tw w t d pt t st ps d w t l tw p t s ll t d l s s l t d d l d t d w t s l p ss ppl t s t s t s t s lts

c e e e

s s s d d S 6 S d t

f nc s

a ang ar N ra N wor or I ran on rc an S S 0

arr r an J No D cr ar N ra N wor rc c r ca on an R a a on R or No NS R 0 c n c n r nc n No r 0

a Ro a NN n r a ac n ar rc c r NN

n J S an SI gn o g a ar N ra N wor or ag roc ng J o a o n ca on an I ag R r n a on o No

aran r Ro a S o ga n a g a arc c r n ng NN n r a ac n roc o I n or o on ar N ra N wor an r ca on on on r

aran r Ro a S o ga S an a g a NN arc c r gn n w r c ron c rc an S I In rna ona on r nc on or ga

Nag S o ga on g ra a r NN a or on G ar N ra N wor an r ca on 00 NN 00 roc ng o 00 I In rna ona or o on ag J 00 xan r on o o a ca II a a a D na co N D

A Binary Multiplier Using RTD Based Threshold Logic Gates

PM Kelly[1], CJ Thompson[1], TM McGinnity[1], LP Maguire[1]

[1] Intelligent Systems Engineering Laboratory, Faculty of Informatics, University of Ulster
Derry, Northern Ireland, BT48 7JL, UK.
Phone: +44-28-7137-5293, Fax: +44-28-7137-5570
email: pm.kelly@ulster.ac.uk

Abstract. A binary multiplier implemented using RTD based threshold logic gates is presented. The circuit demonstrates how small-scale threshold logic gates implementing standard boolean functions can be used to replace conventional boolean gates and achieve reduced circuit complexity. The performance of the gates and multiplier are simulated in HSPICE and the results are presented.

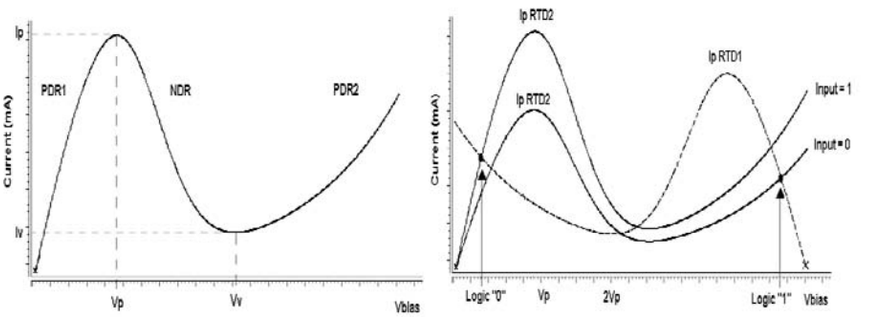
1 Introduction

Threshold Logic Gates (TLGs) are conceptually similar to the early McCulloch-Pitts (MCP) model of the neuron and are normally used to implement linearly separable binary functions [1]. Circuit designs employing TLGs set out to achieve greater functional density resulting in less complex circuits. Given scalable devices this characteristic presents an interesting possibility for the future of VLSI with respect to Moore's Law. Normally, reduced feature sizes of transistors is the means by which increased functional density and capacity in integrated circuits are achieved. However, threshold logic circuit design offers a complimentary technique which when taken in conjunction with reduced feature sizes could further enhance the functional density. If we also make use of technologies, devices and circuits that increase functional density a further enhancement is possible. An example of these are RTD-based circuits including TLGs which have been the subject of investigation by a number of researchers [2,3,4]. These devices and circuits demonstrate an inherent increased functionality due to the negative differential resistance (NDR) characteristics and the effect of the resulting current-voltage inversion. RTD based TLGs were published and patented in the 1990s and have been implemented in circuit designs demonstrating lower complexity with increased functionality as well as robust high-speed performance [5,6,7]. Circuits employing RTDs and using a modified threshold logic gate approach have also been shown to successfully perform non-linearly separable functions [8]. In this paper a range of threshold logic gates, implementing the basic Boolean functions, are used in the design of a binary multiplier. The gate designs use models of manufactured InP based RTDs and HFETs. The circuits were simulated using HSPICE .

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 41-48, 2003.
c

2 RTD Characteristics

The fundamental circuit component used in the construction of the TLGs presented in this paper is the RTD. The current voltage characteristics of RTDs display negative differential resistance due to the resonant tunneling effect that determines their behaviour. A typical I/V characteristic is shown in figure 1(a). This shows that a single device has three main regions: a positive-differential-resistance region (PDR1), a negative-differential-resistance region (NDR) and a second positive-differential-resistance region (PDR2). The I-V curve demonstrates that the device has two positive stable states, one in PDR1 and one in PDR2. The simulated devices used in this research have characteristics that display a peak current when the bias voltage is 0.2 volts. The valley current occurs when the bias voltage is increased to 0.55 volts. When the devices are deployed in the TLGs the resulting high and low logic levels can be latched with a holding current of approximately $220\mu A$.



(a) (b)

Fig. 1. (a) Typical I-V characteristic curve for a RTD (b) Load line depiction of a RTD series pair operation

By connecting two RTDs in series and then applying a clocked bias to the series chain, one of the RTDs can latch to PDR2. The diagram in figure 1(b) uses the load line of RTD1 to illustrate this. The RTD with the lowest peak current will latch to the stable point in its PDR2 region. The other RTD returns to the stable point in its PDR1 region. By choosing Vck so that it is slightly higher than the voltage of the stable point in PDR2 only one of the RTDs at a time will be able to develop enough voltage to take it to its PDR2 region. The operation of this fundamental series pair allows the development of binary threshold logic circuits. The underlying principle of operation described here is known as monostable-bistable transition logic element (MOBILE), which derives from the Goto RTD pair and is a well documented technique employing the latching capability of the RTD [9,10].

3 Threshold Gate Operation

The diagram shown in figure 2 illustrates the principle of operation for threshold logic gates. The principles are similar to the early McCulloch-Pitts (MCP) model of an artificial neuron [1]. Both positive and negative weighted inputs *XiWi* are used to achieve linearly separable boolean functions. Equation (1) gives the conditions for switching of the output.

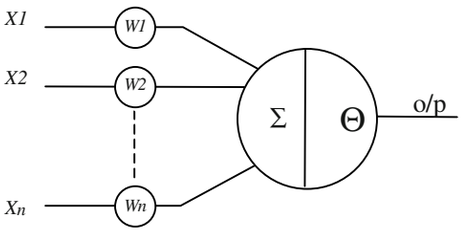


Fig. 2. McCulloch-Pitts neuron-threshold logic gate concept

$$o/p = 1 \text{ iff } \sum_i^n XiWi \geq \Theta \text{ otherwise } o/p = 0 \quad (1)$$

The RTD circuits used here to achieve threshold logic gates are given in figures 3 and 4 and are based on circuit topologies using the MOBILE configuration.

4 Multiplier Circuit

The multiplier circuit comprises a number of different threshold logic gates implementing basic boolean functions. The gates are then combined in a circuit topology based on standard boolean logic equations. This approach allows the use of the standard design techniques to arrive at a solution to the required boolean function. As the circuits are inherently less complex there is a reduction in transistor count for the implementation presented here compared to a multiplier using standard CMOS based logic gates.

The fundamental threshold logic gates used within the multiplier are an AND gate, an XOR gate, a buffer gate and a gate implementing the AND function with one of its inputs inverted. The circuit diagrams and performance data for each of the gates are presented in sections 4.1 and 4.2.

4.1 AND Gate and Buffer Gate

The circuit layouts for the RTD/HFET AND gate and buffer gate are given in figure 3. To realise the AND function RTD areas are selected so that the combined currents of all the RTDs above the summing node is only greater than the total current below the summing node when both inputs X1 and X2 are high.

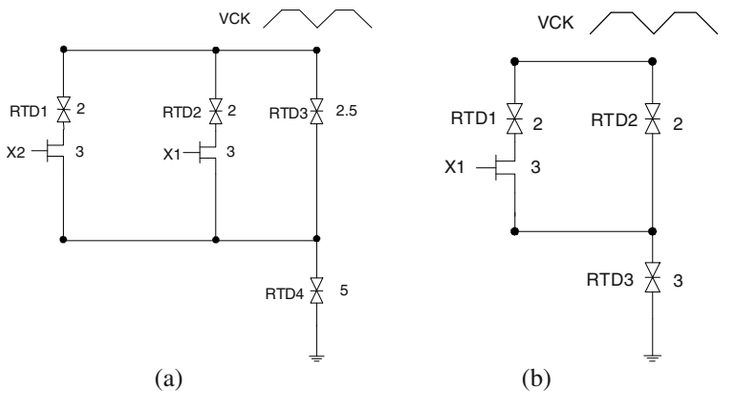


Fig. 3. (a) Threshold logic AND gate (b) Buffer Gate

RTD areas shown in figure 3 are in μm^2 and the peak current density of the devices is 21kA/cm^2. The input branches to both circuits use depletion type HFET's (0.25µm gate length) with width to length (W:L) ratios as shown in figure 3. Performance data for both circuits is given in table 1. The buffer circuit illustrated in figure 3(b) is a simplified version of the AND gate using only one input branch. The buffer circuit simply generates an output which reflects the circuit input i.e. when the input is high the output is high and when the input is low the output is low. The purpose of the buffer is to ensure that all the output signals are available at the appropriate clock phase.

4.2 XOR gate

The XOR gate designed by the authors is illustrated in figure 4(a). It is an adaption of the MOBILE configuration with the use of both positive and negative weighted input sections. The XOR function is realised by placing a series connected HFET pair in parallel with the driver RTD. Using this configuration both inputs must be high to direct current away from the summing node ensuring that the circuit output is logic low when both inputs are high. In the XOR gate depletion type HFETs (0.25µm gate length) are used above the summing node with enhancement mode devices (0.25µm gate length) used in the series connected pair below the summing node. W:L ratios for all devices in both circuits are given in figure 4.

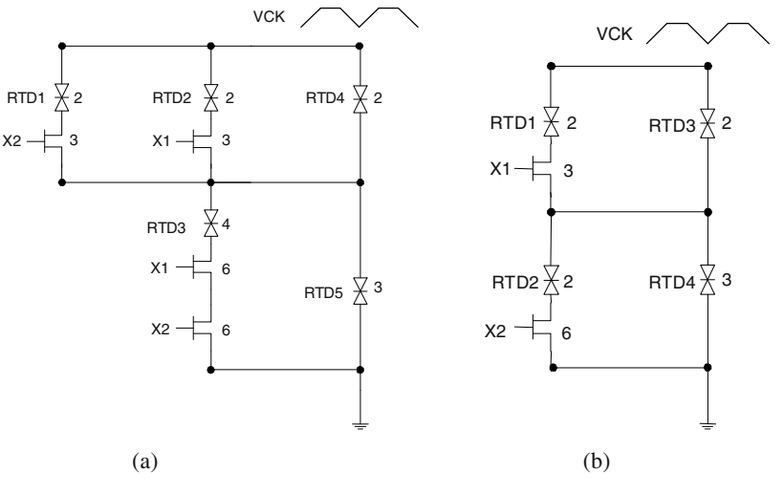


Fig. 4. (a) Threshold logic XOR gate (b) Threshold logic gate for function $X1.\overline{X2}$

RTD areas given in figure 4 for both circuits are given in μm^2 and the peak current density of the devices is 21kA/cm^2. Performance data for both circuits is given in table 1. The threshold logic gate shown in figure 4 (b) uses both positive and negative weighted inputs. The gate is designed to output a logic high when X1 is high and X2 is low ($X1.\overline{X2}$), and output logic low for all other input combinations. As with the XOR gate depletion mode HFETs (0.25μm gate length) are used in the positive weighted section with enhancement mode HFETs (0.25μm gate length) in the negative section.

Table 1. Performance data for the XOR gate and the $f = X1.\overline{X2}$ gate.

Circuit	Clock	Logic high	Logic low	Rise time	Delay	Average Power	Max Power
AND	0.75V	0.701V	0.074V	150pS	90pS	167μW	559μW
Buffer	0.75V	0.707V	0.0711V	160pS	90pS	95μW	338μW
XOR	0.75V	0.707V	0.101V	160pS	90pS	114μW	665μW
$X1.\overline{X2}$	0.75V	0.707V	0.071V	150pS	90pS	103μW	343μW

4.3 Multiplier design

The multiplier circuit block diagram is shown in figure 5. The purpose of the diagram is to show that all of the functions of the individual gates are achieved using small (two inputs) threshold logic gates, these are then combined to produce the required binary function. The block diagram shown in figure 5 was the template for the RTD/HFET circuit designs that follow in figures 6 and 7. The circuit layout for the multiplier is given in individual clock stages for clarity and is shown in figures 6 and 7 respectively.

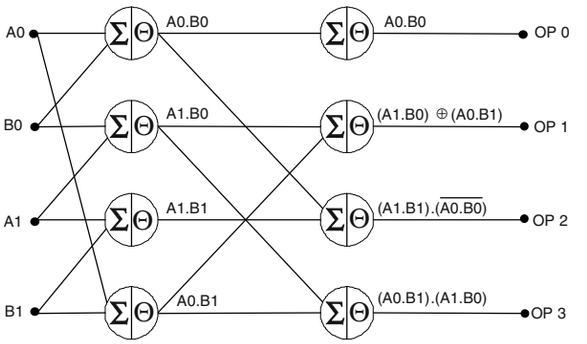


Fig. 5. Block diagram of the binary multiplier showing the function of each of the TLGs. Two two-bit binary numbers A1A0 and B1B0 are multiplied and the four-bit binary number OP3OP2OP1OP0 is output

Simulation results are presented in figure 8. Performance data for the complete circuit is given in table 2. The multiplier circuit was assessed for robustness in terms of clock variation and device area variation. The circuit displayed correct functionality under load conditions for a clock voltage variation of –15% and also withstood RTD area variations of $\pm 10\%$.

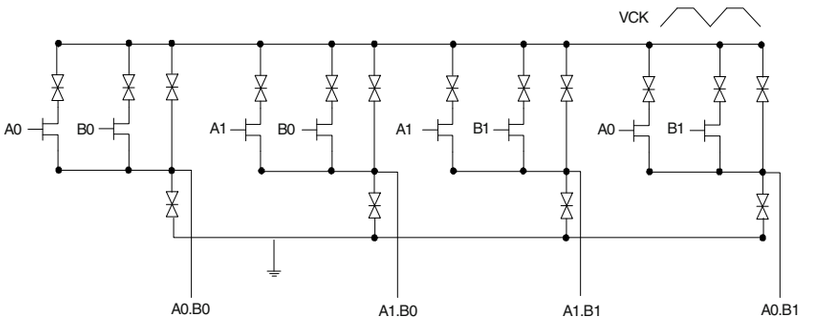


Fig. 6. RTD/HFET circuit design for the first clock phase

The two circuits shown operate on a two-clock system where the first phase of multiplication is evaluated on the leading edge of the first clock. The second phase is evaluated on the leading edge of the second clock, which overlaps the negative going edge of the first clock. This is necessary to ensure that the second stage circuit has the correct values at its inputs when the second clock leading edge arrives. The multiplier showed correct operation for all of the 16 possible inputs applied.

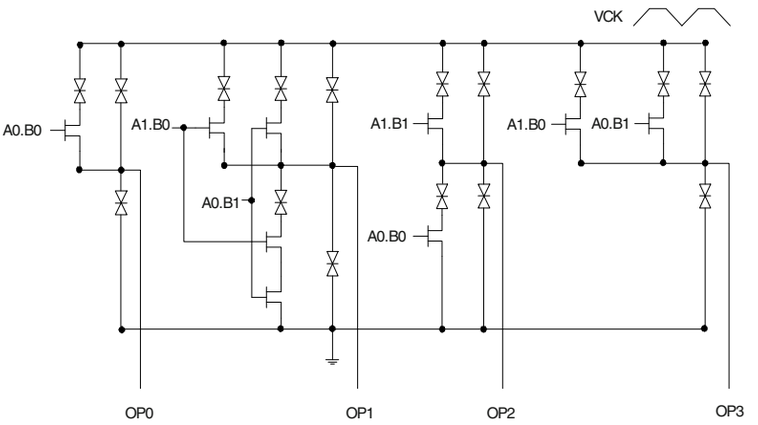


Fig. 7. RTD/HFET circuit design for the second clock phase

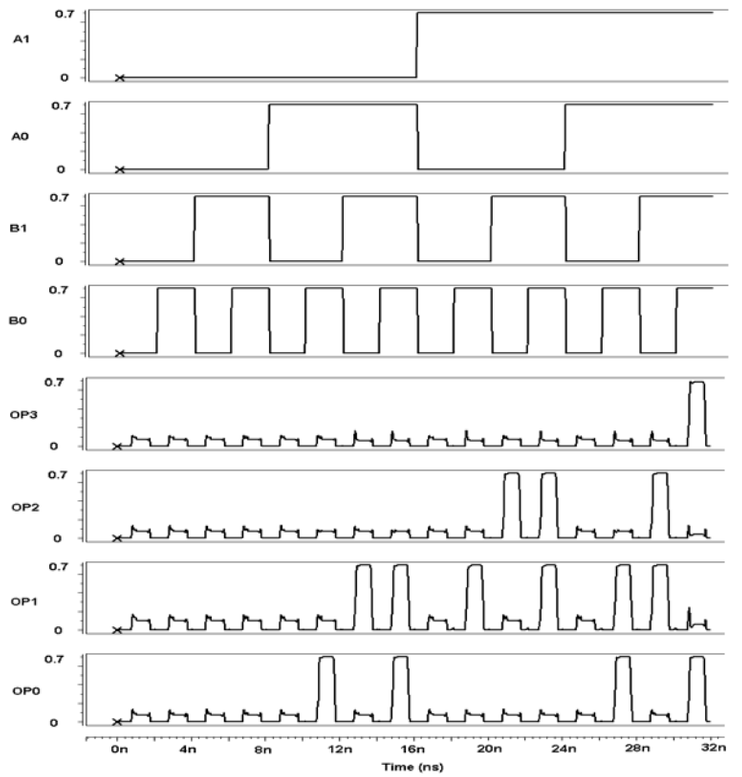


Fig. 8. Output waveforms from the HSPICE simulation of the multiplier

Table 2. Performance data for the multiplier circuit

Clock	Logic high	Logic low	Rise time OP0	Rise time OP1	Rise time OP2	Rise time OP3	Delay	Avg Power	Max power
0.75V	0.68V	0.1V	150pS	170pS	150pS	150pS	95pS	1mW	2.98mW

5 Summary

An RTD based binary multiplier circuit design was presented that demonstrated the use of threshold logic gates as a means of achieving important processor operations such as multiplication. All of the circuits presented were simulated and found to exhibit correct functionality and a robust circuit operation tolerant of device parameter and clock voltage variation. The inherent low complexity of RTD based threshold logic gates allows for gains in functional density. Using two input RTD/HFET threshold logic gates allowed conventional boolean design methods to be applied whilst still increasing the functional density through the low complexity of the circuits.

Acknowledgements

The authors wish to thank: Christian Pacha, Infineon Corporate Research Munich, Germany and Werner Prost, University of Duisburg, Germany. The investigation was funded by the EU QUDOS project IST 2001-32358

References

1. W.S. McCulloch, W. Pitts “A logical calculus of the ideas immanent in nervous activity”, Bulletin of Mathematical Biophysics, vol. 5, pp 115-133, 1943.
2. C. Pacha, U. Auer, C. Burwick, P. Glösekötter, A. Brennemann, W. Prost, F-J. Tegude, and K. F. Goser, “Threshold Logic Circuit Design of Parallel Adders Using Resonant Tunnelling Diodes,” IEEE Trans. on Very Large Scale Integration (VLSI) Systems, Vol.8, No. 5, Oct. 2000.
3. A. Seabaugh, B. Brar, T. Broekaert, G. Frazier, F. Morris, P. van der wagt, and E. Beam III, “Resonant tunneling Circuit Technology: Has it Arrived?” Gallium Arsenide Integrated Circuit (GaAs IC) Symposium, 1997. Technical Digest 1997, 19th Annual, 15-17 Oct 1997 Page(s): 119 –122.
4. P. Mazumder, S. Kulkarni, M. Bhattacharya, J. P.Sun, and G. I. Haddad, “Digital Circuit Applications of Resonant Tunneling Devices,” *Proc. IEEE*, Vol. 86, No. 4, April 1998.
5. W. Prost et al., “LOCOM” EU IST report No. 28844, Dec. 2000.
6. W. Prost, U. Auer, F-J. Tegude, C. Pacha, K. F. Goser, G. Janssen and T. van der Roer, “Manufacturability and Robust Design of Nanelectronic Logic Circuits Based on Resonant Tunnelling Diodes,” Int. J. Circ. Theor. Appl. 2000; 28:537-552.
7. G Frazier, A Taddiken, A Seabaugh, J Randall. “Nanoelectronic circuits using resonant tunneling transistors and diodes”. 1993 IEEE International Solid-State Circuits Conference. Digest of Technical Papers. ISSCC. IEEE. 1993, pp.174-5. New York, NY, USA.
8. P.M. Kelly, C.J. Thompson, T.M. McGinnity, and L.P. Maguire, “Investigation of a Programmable Threshold Logic Gate Array,” IEEE International Conference Electronics Circuits and Systems, proceedings Vol. II, pp 673-67, Sept 2002.
9. K.J. Chen, K. Maezawa, and M. Yamamoto, “InP-Based High-Performance Monostable-Bistable Transition Logic Element (MOBILE): an Intelligent Logic Gate Featuring Weighted-Sum Threshold Operations,” Jpn. J. Appl. Phys. Vol. 35 (1996) pp. 1172-1177 Part 1, No.2B, February 1996.
10. E. Goto et. al. IRE Transactions on Electronic Computers, EC-9, 25 1960.

Split-Precharge Differential Noise-Immune Threshold Logic Gate (SPD-NTL)

Suryanarayana Tatapudi, and Valeriu Beiu [1]

School of Electrical Engineer and Computer Science, Washington State University
Pullman, Washington 99164-2752, USA
{statapud, vbeiu}@eecs.wsu.edu

Abstract. After a short review of the state-of-the-art, a new low-power differential threshold logic gate is introduced: *split-precharge differential noise-immune threshold logic* (SPD-NTL). It is based on combining the *split-level precharge differential logic*, with a technique for enhancing the noise immunity of threshold logic gates: *noise suppression logic*. Another idea included in the design of the SPD-NTL gates is the use of two threshold logic banks implementing *f* and *f_bar*, and working together with the *noise suppression logic* blocks for enhanced performances. Simulations in 0.25 μm CMOS @ 2.5 V show the functionality of the gate up to 2 GHz. An advanced layout based on high matching centroid techniques is currently under development.

1 Introduction

Research on neural networks (NNs) started sixty years ago. The seminal year for the development of the "science of mind" was 1943 when the article *A Logical Calculus of the Ideas Immanent in Nervous Activity* by McCulloch and Pitts was published [29]. For modeling a neuron, they introduced the threshold logic (TL) gate (TLG):

$$f(x_1, \ldots, x_n) = \text{sgn}\left(\sum_{i=1}^{n} w_i x_i - \theta\right) \quad (1)$$

where w_i is the synaptic *weight* associated to x_i, θ is the *threshold*, and n is the *fan-in*. The general belief that a neuron is a TLG can be questionable. That is why, the TL model has been tested on a spike train generated by the Hodgkin-Huxley model with a stochastic input [22]. The result was that the TL model correctly predicts nearly 90% of the spikes, justifying the description of a neuron as a TLG.

The tremendous impetuous of VLSI technology has made neurocomputer design a lively research topic. There are many theoretical complexity results showing that TL circuits (TLCs) are more powerful than classical Boolean circuits [9]. Beside, TLGs

[1] V. Beiu is partly sponsored by the Air Force Research Laboratory under agreement number F29601-02-2-0299. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory or the U.S. Government.

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 49-56, 2003.

(or their variations) have been used in MIPS R2010 [20], SUN Sparc V9 [27], a CMOS fingerprint sensor array [21], and very recently in the Itanium 2 microprocessor [30]. Finally, the emerging devices (*e.g.*, resonant tunnelling, single electron, etc.) have been used for quite some time for implementing TLGs and TLCs. All of these have led to many different TLG implementations (see [10, 3, 4]).

In this paper we shall focus only on differential TLGs. After a brief review of the state-of-the-art (for details see [8]), we will present a new low-power differential TLG called *split-precharge differential noise-immune threshold logic* (SPD-NTL). It is based on combining the *split-level precharge differential logic* SPDL [26], with a technique for enhancing the noise immunity of TLGs [2, 5, 6], known as *noise suppression logic* (NSL). Another idea included in the design of SPD-NTL gates is the use of two TL banks implementing *f* and *f_bar* [7]. This technique works very well together with the NSL enhancing the performances of the TLG.

2 Differential Implementations of Threshold Logic Gates

Energy efficiency design has been the driving force behind a plethora of differential gate designs. They can achieve very low power consumption levels, while operating at very high speeds. They can also be easily modified to work asynchronously (based on done-enable signals). Here is a long list of differential Boolean gates: differential cascode voltage switch DCVS (1984), differential split-level logic DSLL (1985), sample-set differential logic SSDL (1986), differential pass-transistor logic DPTL (1987), enable/disable CMOS differential logic ECDL (1988), latched CMOS differential logic LCDL (1991), differential current switch logic DCSL (1996), charge recycling differential logic CRDL (1996), half-rail differential logic HRDL (1997), current sensing differential logic CSDL (1998), asynchronous sense differential logic ASDL (1999), no-race charge-recycling differential logic NCDL (1999), and split-level precharge differential logic SPDL (2001). A similar list for TL includes: cross-coupled inverters with asymmetrical loads CIAL (1995), latch-type threshold logic LCTL (1995), cross-coupled inverters with asymmetrical loads threshold logic CIALTL (1998), dynamic latched sense amplifier (1998), single input current-sensing differential logic SCSDL (1999), CMOS capacitor coupling logic C^3L (2000), balanced capacitive threshold logic B-CTL (2000), current-mode threshold logic CMTL (2000), discharge CMTL (2000), equalized CMTL (2000), differential current-switch threshold logic DCSTL (2001), charge recycling threshold logic CRTL (2001), and self-timed threshold logic STTL (2002). We shall briefly review these here, while the interested reader should consult [8, 10].

Two basic approaches for implementing TLGs are: capacitive and conductance. The concept underlying capacitive TLGs is the use of an array of capacitors to implement the weighted sum of inputs. The idea was introduced as early as 1966 [12]. Capacitive TLGs can be classified into: *capacitive threshold logic* (CTL), and *neuron MOS* (νMOS). A few comparisons [31, 32, 13], draw the following conclusions:

- the νMOS operation is simpler than that of the CTL;
- the maximum attainable *fan-in* by νMOS is an order of magnitude less than that of CTL gate;

- the delay has a logarithmic dependence with respect to large *fan-ins* (*fan-in* ≤ 255 in [31], *fan-in* ≤ 64 in [32]), while for small *fan-ins* (*fan-in* ≤ 20 [13]) the behaviour looks linear: $1 + 0.35n$ (where n is the *fan-in*).

The idea of using switched capacitors, switches, and inverters, and taking advantage of the inherent saturation of the inverters to implement the perceptron nonlinearity, was originally introduced in 1987 [39]. This first approach required a somehow complex three-phase clock. It has quickly evolved into a simpler two-phase clock solution [31]: the *capacitive threshold logic* (CTL). This has large *fan-in* capability (up to 255), but also large delays and area, and DC power consumption.

- A differential version is the *balanced-CTL* (B-CTL) [17]. The requirement for a highly precise reference voltage is eliminated here by implementing functions with *thresholds* equal to 0. Two banks of capacitors connected to a differential amplifier form the basic structure, with one additional half-capacitor unbalancing the voltage level. B-CTL gates are reported to be faster than CIAL gates [36].

Neuron MOS TLGs are based on an idea introduced in the mid 60s [12]. It was rediscovered in 1991 [37]. The *static* νMOS TLG is very simple and compact, but has DC power consumption. This static power can be eliminated and the speed increased by a current comparison between a νMOS transistor and a reference device, using a positive feedback circuit.

- One configuration is the *sense-amplifier* νMOS TL [24]. It employs a current-controlled latch-sense amplifier circuit. Variations can be found in [40]. This solution is similar to the digital comparator from [28]. Speed improvements of 5×, and power savings over the static νMOS were reported [40].
- Another variation, *CMOS capacitor coupling logic* (C^3L), uses the capacitor coupling technique and a current sense amplifier [19]. Fluctuations of the device parameters are compensated by the differential configuration.
- The *charge recycling threshold logic* (CRTL) gate introduced in [14] is based on CRDL [23]. CRTL gates exhibit high speed (even for high *fan-ins*), while also having low power consumption. CRTL gates achieve the highest speed and 15-20% lower power consumption when compared with clocked νMOS [24], C^3L [19], and LCTL [1].
- A *self-timed threshold logic* (STTL) has been proposed in [15]. The gate is based on a cross-coupled nMOS transistor pair. The enable signals are passed to the next stage, being propagated in a self-timed fashion. The solution is low power, and eliminates the clock at the expense of a double-rail signalling and the additional "enable generate" block.

The other class of differential TLG implementations is the current/conductance category. Two parallel-connected banks of nMOS transistors are used for implementing the weighting operation, followed by a current CMOS comparator for the threshold operation.

- The operation of *cross-coupled inverters with asymmetrical loads* (CIAL) was exploited to implement digital (bus) comparators [28], a particular TLG.
- A generic *latch-type TL* (LCTL) gate was proposed in [1]. Its consists of a CMOS current-controlled latch providing both the output and its complement, and two input arrays having an equal number of parallel transistors whose gates are the inputs of the TLG. Two extra transistors guarantee correct operation for

the case when the weighted sum of inputs is equal to the *threshold* value. Current flows only during transitions.

- The speed performance of LCTL was improved in [36], where the nMOS banks are external to the latch. It is called *cross-couple inverters with asymmetrical loads threshold logic* (CIALTL), but it is different from CIAL [28].
- The circuit arrangement for realizing logic elements that can be represented by *threshold* value equations patented by Prange *et al.* [35] is a simplified CIAL.
- *Single input current-sensing differential logic* (SCSDL) [38] is based on CSDL [34]. Yield analysis for SCSDL in 0.35 μm CMOS has showed that *fan-in* ≤ 14.
- *Differential current-switch threshold logic* (DCSTL) [33] is based on DCSL. It restricts the voltage swing of the internal nodes for lowering the power consumption. Reported experiments show that DCSTL exhibits better power-delay product than: LCTL [1] and CIALTL [36].
- *Current-mode threshold logic* (CMTL) [11] achieves low power by limiting the voltage swing on interconnects and on the internal nodes. Various clocked cross-coupled loads have led to discharged CMTL (DCMTL) and equalized CMTL (ECMTL).

A differential TLG bridging the gap between capacitive and conductance implementations has also been proposed [16]. The key computational concept is to use a floating-gate device as a programmable-switched conductance. Two parallel Flash-EEPROM banks implement the weighted sum of inputs with positive *weights*, and the weighted sum of inputs with negative *weights*. The rest of the circuit measures the conductance based on the current through the 'memory' cells.

All the TLGs based on current comparisons are relatively sensitive to noise and mismatch of process parameters. Reliability can be improved by known layout and circuits techniques where the devices behaviour is matched (substrate voltage control, shield and isolations, centroid layout) for reducing statistical parameter variations.

3 A New Differential Threshold Logic Gate

All the differential TL solutions presented compare the sum of *weights* with a *threshold* [14, 15, 19, 24, 35, 38] (eventually unbalanced as in [33]), or compare two weighted sums [1, 16, 17, 28, 36] corresponding to the positive and the negative *weights*. The functions implemented in this second case have *threshold* 0. Additional transistors are needed to differentiate the case when the two weighted sums are equal.

A quite simple but efficient idea is to implement the function *f* with one TL bank, while implementing *f_bar* with the other TL bank. It is well known that if *f* is a TL function, *f_bar* is a TL function having the same weights w_i if the inputs x_i are inverted (x_i*_bar*), and the *threshold* is changed. The fact that *f* and *f_bar* always have transitions in opposite directions leads to increased speed, and also to better noise margins [7]. The noise margins can be enhanced even more—while also marginally increasing the speed—by increasing the gap using a non-linear solution both for *f* and *f_bar*, like *e.g.*, NSL [2, 5, 6]. The resulting solution is noise immune by design (and also very fast). That is why we call it *noise-immune threshold logic* (NTL). It can be

used in conjunction with any differential implementation. In this paper we will show how NTL works in conjunction with SPDL [26].

SPDL is an improvement on other charge recycling differential logic (*e.g.*, HRDL, CRDL) in terms of power dissipation, propagation delay, increased reliability, avoiding metastable states, and large *fan-out*. The charge recycling makes the outputs switch from $V_{DD}/2$ to either V_{DD} or GND during the comparison phase. The decrease in power dissipation is achieved by not turning on the pull-up and pull-down transistors at the same time. SPDL can also be self-timed (asynchronous), therefore reducing the power consumed at the chip level by eliminating the (complex) clock distribution.

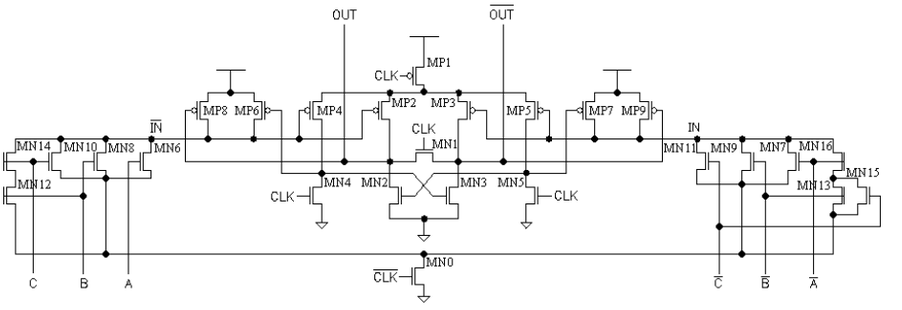


Fig. 1. Schematic of a 3-input SPD-NTL gate implementing $A \vee (B \wedge C) = \mathrm{sgn}(2A+B+C-1.5)$.

The SPDL part is represented by MN0-MN5 and MP1-MP9 (see [26]). The version used here is the clocked one. The left bank computing IN_*bar* = *f*_*bar* is formed by MN6, MN8, MN10, which are driven by A, B, C. The right bank computing IN = *f* is formed by MN7, MN9, MN11, which are driven by A_*bar*, B_*bar*, C_*bar*. Additional NSL structures are used for increasing the 'gap' (therefore enhancing the noise immunity of the gate). The NSL for IN_*bar* is formed by MN12 and MN14 [2, 5, 6], while the NSL for IN is formed by MN13, MN15, and MN16. The proper sizing of the transistors MN6-MN11 for encoding the *weights* associated to the inputs, makes it that the W/L of MN6 is double that of MN8, and MN10. The same is true for the right bank, where the W/L of MN7 is double that of MN9, and MN11. The NSL logic blocks (MN12-MN14 and MN13-MN15-MN16) have to be sized at least as large as MN6 (making them larger will always improve on the noise margins [6]).

A straightforward layout of the gate was done in CMOS 0.25 μm @ 2.5 V. All the post-layout simulations have been at 100 ºC, with the inputs varying from 0.25 V to 2.25 V, and with a load of eight minimum inverters on OUT, and eight minimum inverters on OUT_*bar*. The gate has been tested at the following four clock frequencies: 100 MHz, 500 MHz, 1 GHz, and 2 GHz. The average current when running continuously was: 271 μA, 523 μA, 736 μA, and 924 μA. The delay of the gate for the given conditions is around 150 ps, leading to a power-delay-product of about: 100 fJ @ 100 MHz, 200 fJ @ 500 MHz, 280 fJ @ 1 GHz, 350 fJ @ 2 GHz. The simulations for the 2 GHz case are presented in Fig. 2.

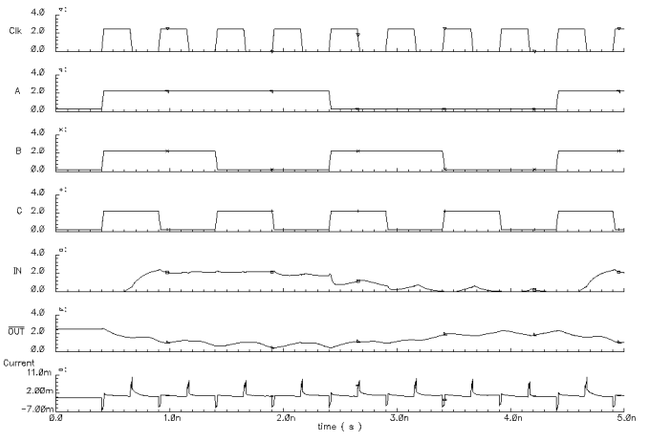


Fig. 2. Simulation results for the gate running at 2 GHz.

Finally, another idea is to systematically use centroid techniques. 4, 8, and even 16 transistors will replace each transistor. A 4-transitor structure is depicted in Fig. 3 (a), while Fig. 3 (b) presents the 8-transistor structure for high matching from [25]. For even better matching we have designed a 16-transistor structure (Fig. 3 (c)). This is formed of two 8-transistor structures one inside the other (imbricated), and rotated by 45º relative to one another. The complex layout is in progress (Fig. 4).

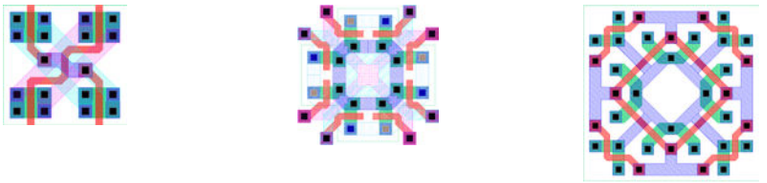


Fig. 3. Centroid layout structures for better matching: (a) classical 4-transistor structure; (b) advanced 8-transistors structure; (c) 16-transistors structure for ultra-high matching (introduced in this article).

4 Conclusions

The present state-of-the-art of shows a large variety of differential TLGs. Some of these are quite advanced (*e.g.*, asynchronous [4, 15, 18, 26]). They allow for drastical reductions of the dissipated power (when compared to earlier implementations [10]). A novel differential TLG has also been proposed. It incorporates two new ideas: using *f* and *f_bar*, and adding nonlinear data-dependent terms (NSL). Centroid techniques are currently being investigated for ultra-high matching. Fast and very low-power differential TLGs are therefore implementable, the major differences between one solution and another being the power-delay tradeoffs.

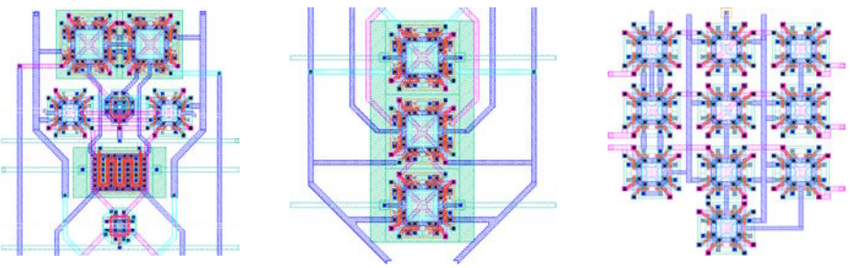


Fig. 4. Systematic centroid layout of the SPD-NTL gate: (a) the SPDL differential; (b) the SPDL pull-up structure; (c) the two nMOS banks including the two NSL structures.

References

1. Avellido, M.J., Quintana, J.M., Rueda, A., Jiménez, E.: A low-power CMOS threshold-gate. Electron. Lett. 31 (1995) 2157–2159
2. Beiu, V.: Ultra-fast noise immune CMOS threshold gates. Proc. MWSCAS'2000 vol. 3 (2000) 1310–1313.
3. Beiu, V.: Adder and multiplier circuits employing logic gates having discrete, weighted inputs and methods of performing combinatorial operations therewith. U.S. Patent 6 205 458 (Mar. 20, 2001)
4. Beiu, V.: Logic gate having reduced power dissipation and method of operation thereof. U.S. Patent 6 259 275 (Jul. 10, 2001)
5. Bieu, V: On higher order noise immune perceptrons. Proc. IJCNN'2001 vol. 1 (2001) 246–251
6. Beiu, V.: Noise tolerant conductance-based logic gate and methods of operation and manufacturing thereof. U.S. Patent 6 430 585 (Aug. 6, 2002)
7. Beiu, V.: Low-power differential conductance-based logic gate and method of operation thereof. U.S. Patent allowed US/09/667 312 (Feb. 17, 2003)
8. Beiu, V., Quintana, J.M., Avedillo, M.J.: Review of differential threshold gate implementations. Proc. NCI'03 (May 2003)
9. Beiu, V.: On perceptron circuit complexity results and some practical applications. Proc. IJCNN'03 (Jul. 2003)
10. Beiu, V., Quintana, J.M., Avedillo, M.J.: VLSI implementations of threshold logic: a comprehensive survey. Accepted IEEE Trans. Neural Networks (Sep. 2003)
11. Bobba, S., Hajj, I.N.: Current-mode threshold logic gates. Proc. ICCD'2000 (2000) 235–240
12. Burns, J.R., Trenton, N.J., Powlus, R.A.: Threshold circuit utilizing field effect transistiors. U.S. Patent 3 260 863 (Jul. 12, 1966)
13. Celinski, P., Al-Sarawi, S., Abbott, D.: Delay analysis of neuron-MOS and capacitive threshold-logic. Proc. ICECS'2000 vol. 2 (2000) 932–935
14. Celinski, P., López, J.F., Al-Sarawi, S., Abbott, D.: Low power, high speed, charge recycling CMOS threshold logic gate. Electron. Lett. 37 (2001) 1067–1069
15. PCelinski, P., López, J.F., Al-Sarawi, S., Abbott, D.: Compact parallel (m, n) counters based on self-timed threshold logic. Electron. Lett. 38 (2002) 633–635
16. Fabbrizio, V., Raynal, F., Mariaud, X., Kramer, A., Colli, G.: Low-power, low-voltage conductance mode CMOS analog neuron. Proc. MicroNeuro'96 (1996) 111–115

17. García, J.L., Ramos, J.F., Bohórquez, A.G.: A balanced capacitive threshold logic gate. Proc. DCIS'2000 [available at www.el.uma.es/Ppepefer/DCIS2000.pdf]
18. Gayles, E.S.: Data enabled logic circuits. U.S. Patent 6 078 196 (Jun. 20, 2000)
19. Huang, H.Y., Wang, T.N.: CMOS capacitor coupling logic (C^3L) logic circuits. Proc. APASIC'2000 (2000) 33–36
20. Johnson, M.G.: A symmetric CMOS NOR gate for high-speed applications. IEEE J. Solid-State Circuits 23 (1988) 1233–1236
21. Jung, S., Thewes, R., Scheiter, T., Goser, K.F., Weber, W.: A low-power and high-performance CMOS fingerprint sensing and encoding architecture. IEEE J. Solid-State Circuits 34 (1999) 978–984
22. Kistler, W.M., Gerstner, W., van Hemmen, J.L.: Reduction of the Hodgkin-Huxley equations to a single-variable threshold model. Neural Computation 9 (1997) 1015–1045
23. Kong, B.-S., Choi, J.-S., Lee, S.-J., Lee, K.: Charge recycling differential logic for low-power application. Proc. ISSCC'96 (1996) 302-303
24. Kotani, K., Shibata, T., Imai, M., Ohmi, T.: Clocked-controlled neuron-MOS logic gates. IEEE Trans. Circuits Syst. II 45 (1998) 518–522
25. Lan, M.F., Tammineedi, A., Geiger, R.: A new current mirror layout technique for improved matching characteristics. Proc. MWSCAS'99 (1999) 1126–1129
26. Lee, J., Park, J., Song, B., Kim, W.: Split-level precharge differential logic: a new type of high-speed charge-recycling differential logic. IEEE J. Solid-State Circuits 36 (2001) 1276–1280
27. Lev, L.A.: Fast static cascode logic gate. U.S. Patent 5 438 283 (Aug. 1, 1995)
28. López, J.A.H., Tejero, J.G., Ramos, J.F., Bohórquez, A.G.: New types of digital comparators. Proc. ISCAS'95 vol. 1 (1995) 29–32
29. McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. Bull. Math. Biophysiol. 5 (1943) 115–133
30. Naffziger, S.D., Colon-Bonet, G., Fischer, T., Reidlinger, R., Sullivan, T.J., Grutkowski, T.: The implementation of the Itanium 2 microprocessor. IEEE J. Solid-State Circuits 37 (2002) 1448–1460
31. Özdemir, H., Kepkep, A., Pamir, B., Leblebici, Y., Çilingiroğlu, U.: A capacitive threshold-logic gate. IEEE J. Solid-State Circuits 31 (1996) 1141–1150
32. Pădure, M., Dan, C., Coțofană, S., Bodea, M., Vassiliadis, S.: Capacitive threshold logic: a designer perspective. Proc. CAS'99 vol. 1 (1999) 81–84
33. Pădure, M., Coțofană, S., Dan, C., Bodea, M., Vassiliadis, S.: A new latch-based threshold logic family. Proc. CAS'01 vol. 2 (2001) 531–534
34. Park, J., Lee, J., Kim, W.: Current sensing differential logic: a CMOS logic for high reliability and flexibility. IEEE J. Solid-State Circuits 34 (1999) 904-908
35. Prange, S., Thewes, R., Wohlrab, E., Weber, W.: Circuit arrangement for realizing logic elements that can be represented by threshold value equations. U.S. Patent 5 991 789 (Nov. 23, 1999)
36. Ramos, J.F., López, J.A.H., Martín, M.J., Tejero, J.C., Bohórquez, A.G.: A threshold logic gate based on clocked coupled inverters. Intl. J. Electronics 84 (1998) 371–382
37. Shibata, T., Ohmi, T.: An intelligent MOS transistor featuring gate-level weighted sum and threshold operations. IEDM Technical Digest (1991) 919–922
38. Strandberg, R., Yuan, J.: Single input current-sensing differential logic (SCSDL). Proc. ISCAS'2000 vol. 1 (2000) 764–767
39. Tsividis, Y.P., Anastassiou, D.: Switched-capacitor neural networks. Electron. Lett. 23 (1987) 958–959
40. Weber, W., Prange, S.J., Thewes, R., Wohlrab, E., Luck, A.: On the application of neuron MOS transistor principle for modern VLSI design. IEEE Trans. Electron Devices 43 (1996) 1700–1708

UV-programmable Floating-Gate CMOS Linear Threshold Element "P1N3"

Snorre Aunet[1] and Yngvar Berg[2]

[1] Norwegian University of Science and Technology, Department of Computer and Information Science, Trondheim, Norway,
aunet@ieee.org.
[2] University of Oslo, Department of Informatics,
Oslo, Norway.

Abstract. We present some aspects regarding modelling of floating-gate UV-programmable (FGUVMOS) circuits when used as linear threshold elements, with a simple building block used for generation of Boolean functions as an example. Some comparisons with other floating gate and standard cell CMOS implementations are also done. A new FULL-ADDER structure containing only eight transistors is demonstrated by SPICE simulations, using a power supply voltage of 0.8 V. We argue that the floating-gate linear threshold elements have a significant ultra low-power potential and may allow very simple circuitry, provided that the technology can prove to reach adequate maturity.

1 Introduction

Floating-gate circuits for signal processing have gained an increasing interest during about the last decade, maybe especially due to the neuron MOS concept published in 1991 [1], and an introduction to the field can be found in [2]. Floating-gate UV-programmable CMOS circuits are exploiting standard double-poly CMOS processes like [3] with the goal of making ultra low-power analog and digital [4] as well as neural network circuitry [5], [6].
This paper is organized as follows: FGUVMOS circuits and UV-programming are briefly introduced in section 2, together with a couple of important equations which may be useful for a simple analysis. Section 3 deals with the P1N3 linear threshold element, more specific, and tries to show how a simple 1st order analysis of functioning can be done. Further a circuit combination of two P1N3 elements forming a FULL-ADDER is treated and later demonstrated by SPICE simulations. In section 4 some aspects of the technology like low circuit complexity and ultra low power consumption are briefly mentioned. Section 5 concludes the paper.

2 Floating-Gate UV-programmable Circuits

The circuits may be operated in weak inversion / subthreshold, getting their effective threshold voltages seen from driving nodes altered by a postprocessing technique exploiting UV light. The UV-programming typically lasts 15-30

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 57-64, 2003.

minutes. Selected parts of the chip surface are hit by using metal layers as shields, and by making openings in the pad-layer. In this way UV-activated conductances can be made, like in the left part of figure 1, when a special reverse-biasing scheme is used [4], [7]. Power lines are used in the UV-programming process, and there are virtually no area overhead.

By applying certain combinations of voltages, V_+ and V_-, in programming mode (figure 1), the net charge on the floating gate nodes FGN and FGP can be controlled, and thus the voltages. When the UV-light is switched off the UV-activated conductances diminish, and the charges on the two nodes are determining the efficient threshold voltages seen from the driving nodes under normal operation.

The UV-programming process sets the switching point so that all driven nodes are at $V_{dd}/2$ in the initial operative mode, and establishes the equilibrium drain currents, I_{beq}, typically in the nA to μA range from experience with a 0.6 μm CMOS process [3]. The current levels of a chip might be reprogrammed to different values.

If the sum of the capacitive weights connected between each input and it's

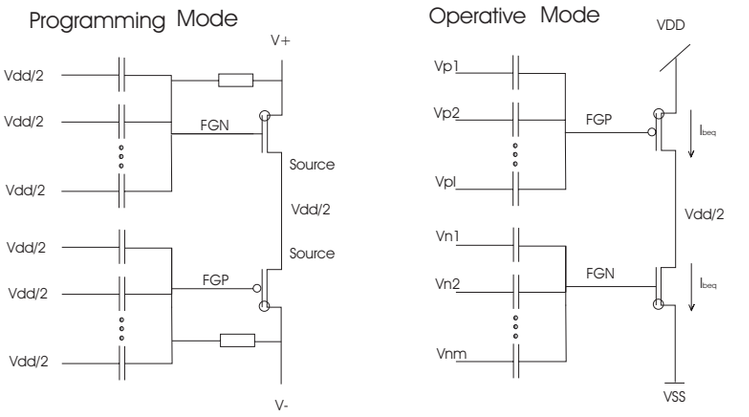


Fig. 1. Schematic of the UV-programming mode and the normal operative mode, based on [7]. The extra circles in the MOSFET symbols indicate UV-conductances. In operative mode the equilibrium current is the drain current at the switching point, when all inputs and the output equal $V_{dd}/2$.

respective floating gate, and the number of drawn capacitances, are equal, the following equations approximate the drain currents for an element like to the right in figure 1 [4]:

$$I_{ds,p} = I_{beq} \prod_{i=1}^{m} exp\{\frac{1}{nU_t}(V_{dd}/2 - V_i)k_i\} \quad (1)$$

$$I_{ds,n} = I_{beq} \prod_{i=1}^{m} exp\{\frac{1}{nU_t}(V_i - V_{dd}/2)k_i\} \quad (2)$$

Here, $k_i = C_i/C_{tot}$ is the capacitive division factor of the ith input capacitor, C_i, and C_{tot} is the total capacitance seen from the floating-gate.
It might be useful being aware that the sum of drawn capacitances is usually less than C_{tot} seen from the floating-gate, so that the input signal of, for example, an inverter is attenuated before reaching the floating-gate [6]. The above equations also suggest a great deal of symmetry between PMOS and NMOS, which is only approximately the truth, but anyway these simple equations have proven useful for making simple behavioural analysis regarding FGUVMOS circuitry. I_{beq} is the balanced equilibrium current, which is the drain current of the transistors when all input signals and driven nodes have a voltage level equalling $V_{dd}/2$. I_{beq} is a function of the effective threshold voltages seen from the driving nodes. For a certain I_{ds} level for PMOS and NMOS there is one set of effective threshold voltages. An FGUVMOS circuit can have different I_{ds} levels for different transistors, due to different layouts, transistor sizes and topologies [6].
A more detailed treatment of CMOS transistors operating in subthreshold can be found in [8].

3 P1N3 linear threshold element

3.1 Simple analysis of P1N3 circuit

When the circuit has been UV-programmed it is "balanced", and the exponentials in the above mentioned equations equal zero. The parts of the equations treated as depending on the input voltages are named e_p and e_n, respectively. To make a simple analysis of the P1N3 circuit we can express them as:

$$e_p = (\frac{V_{dd}}{2} - V_p) \quad (3)$$

$$e_n = (\frac{1}{3}V_x + \frac{1}{3}V_y + \frac{1}{3}V_z - \frac{V_{dd}}{2}). \quad (4)$$

If we consider binary inputs to the linear threshold element, defining a perfect "1" as V_{dd} and a perfect "0" level as V_{ss}, the truth table in figure 3 can be made, depending on the number og 1s in the input vector. If $e_p > e_n$ the output is forced closer to V_{dd} than V_{ss}, and the output is classified as "1". When $e_p < e_n$ a "0" results on the output. From figure 3 it is apparent that the output goes low if, and only if, there are two or more 1s at the inputs. In that case the circuit computes the CARRY' function for binary addition. If e_p is changed from $V_{dd}/2$ the relations between e_p and e_n can change so that the P1N3 element implements other logic functions. If Δe_p is changed to -$2V_{dd}/6$ only one 1 on one of the inputs is sufficient to force the output low, which provides a 3-input NOR function. Apparent from figure 3 is also that Δe_p=$2V_{dd}/6$, right in-between of $Vdd/6$ and $3V_{dd}/6$, will make the P1N3 circuit implement the 3-input NAND function.

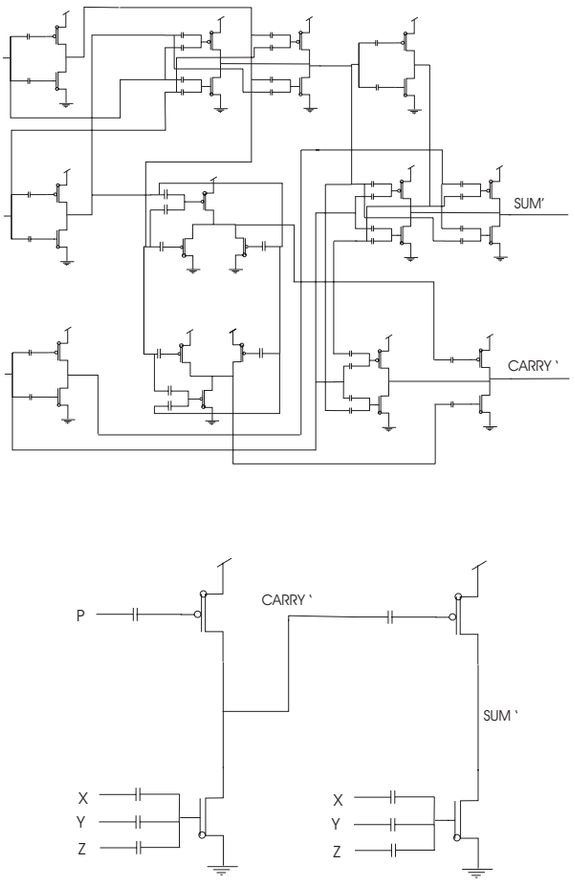


Fig. 2. Alternative schematics for FGUVMOS FULL-ADDER from [4] for producing SUM' and CARRY'. The upper schematic is from an implementation [7] resembling a traditional "transistor as a switch" approach. The lower schematic includes two P1N3 linear threshold elements.

P	number of "1"'s	Δe_p	Δe_n	OUTPUT (CARRY')
$V_{dd}/2$	0	0	-$3V_{dd}/6$	1
$V_{dd}/2$	1	0	-$V_{dd}/6$	1
$V_{dd}/2$	2	0	$V_{dd}/6$	0
$V_{dd}/2$	3	0	$3V_{dd}/6$	0

Fig. 3. The table shows parts of the exponentials, e_p, e_n, and output values, for $V_P = V_{dd}/2$ and different numbers of "1's" on ordinary inputs X,Y,Z. Inputs can be either "0" or "1".

Relative to V_{dd} and V_{ss} a voltage of $V_{dd}/6$ on the input driving the floating gate of the PMOS gives the 3-input NAND function, while $5V_{dd}/6$ corresponds to the 3-input NOR function.

3.2 A combination of two P1N3 elements

Here two P1N3 elements coupled in series, like lowermost in figure 2, are considered. If the previously mentioned C_{tot} is larger than the sum of the drawn capacitances between driving inputs and floating gate of a PMOS or an NMOS, the capacitances may be dimensioned so that driving nodes receives proper voltages to control whether the 3-input NOR or 3-input NAND function will be implemented by a FGUVMOS linear threshold element like P1N3, for example. Then, if the output of the 1st P1N3 element lowermost in figure 2 is high, the 2nd P1N3 circuit need only one high ("1") input to force the output low. This is in accordance with the upper half of the truth table in 4, where the CARRY' node is 1.
If the output of the 1st P1N3 element (figure 4) is low, the 2nd P1N3 circuit need 3 1s on its inputs to produce a 0 on the output.
The truth table of the P1N3 based circuit in figure 2 is presented in figure 4 and may be used to implement the CARRY' and SUM' functionalities for binary addition. If two P1N3 elements, each with all their inputs to floating gates short circuited, are driven by the CARRY' and SUM' nodes respectively, the four P1N3 circuits form a complete FULL-ADDER [6].

number of "1"'s	OUTPUT (CARRY')	SUM'
0	1	1
1	1	0
2	0	1
3	0	0

Fig. 4. The table shows parts of the exponentials, e_p, e_n, and output values, for $V_P = V_{dd}/2$ and different numbers of "1's" on ordinary inputs X,Y,Z. Inputs can be either "0" or "1".

3.3 Spice simulation of four P1N3 elements forming a FULL-ADDER

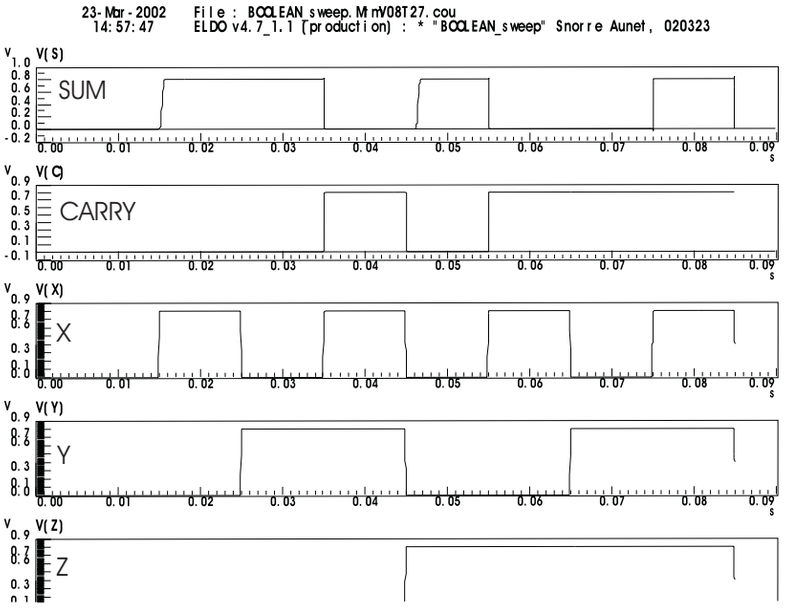


Fig. 5. Simulation for a P1N3 based FULL-ADDER. W/L=20.8μm/1.2μm. Unit capacitances were 70 fF.

A transient simulation of a FULL-ADDER containg four P1N3 elements only, is shown in figure 5. The voltage on the SUM node is about V_{dd} level if, and only if, one or three of the inputs X,Y,Z are at the same, high, level. The voltage on the CARRY node is high ($\approx V_{dd}$) if, and only if, two or three of the binary inputs X,Y,Z are high.

4 Comparisons of FGUVMOS circuits and other CMOS implementations

4.1 Linear threshold elements reducing number of transistors and passive components

From figure 2 it is obvious that the lower schematic using two P1N3 linear threshold elements leads to a significant reduction in the number of transistors, drawn capacitances and wiring. Logic depths for generating SUM' and CARRY'

are also differing, and is summed up in figure 6. The number of basic building blocks for producing Boolean functions can also be significantly reduced by using linear threshold elements compared to the solution from [4]. This makes UV-programming and testing easier, and improves matching of circuitry [5].

# tran.	# tran.	logic depth	logic depth	# different blocks	reference
$CARRY'$	SUM'	$CARRY'$	SUM'		
22	16	5	4	6	[4]
2	4	1	2	2	[9]
2	4	1	2	1	[5], this

Fig. 6. Some comparisons between FGUVMOS FULL-ADDERs (1-bit adder). ("#tran.": number of transistors) [6]

4.2 FGUVMOS ultra low-power potential

An 8-transistor FULL-ADDER might use 2500 times less than a FULL-ADDER using standard cells in the same 0.6μm CMOS technology citeams1 while running at 1 MHz [6], with a power-delay-product of 2.3 fJ [6]. Two other low-power FULL-ADDERs from [10], implemented in another 0.6μm CMOS technology, had PDP numbers of 60 and 92 fJ, while the best among 25 different implementations in a, probably significantly more competitive, 0.35μm technology had a PDP number of 18.4 fJ.

5 Discussion and Conclusion

A simple method for analyzing FGUVMOS linear threshold elements has been shown, together with a new and simple implementation of a FULL-ADDER as a circuit example.
The reduction in transistors, passive components and wiring compared with earlier FGUVMOS [4] as well as standard CMOS solutions may contribute to reduced power dissipation when FGUVMOS circuits are implemented by using linear threshold elements. The weak inversion ("subthreshold") operation also helps in reducing power dissipation simply because of the lower current levels. The possibility to use supply voltages typically in the 200 mV to 800 mV range is also a way of minimizing power consumption, since voltage reduction offers the most direct and dramatic means of minimizing power consumption according to [12]. Circuits oeprating in weak inversion have been shown to consume orders of magnitude less power than the regular strong-inversion circuit at the same operating frequency [13], which seems to be in accordance with the FGUVMOS result of 2500 times less power dissipation of a FGUVMOS FULL-ADDER exploiting linear threshold elements, mentioned above.

A fan-in of two is optimum for building arbitrary neural networks from linear threshold elements [14], and a fan-in of 6-9 can lead to VLSI optimal solutions [15]. Such fan-ins are probably within reach for FGUVMOS technology [6]. If the FGUVMOS technology can prove to be mature enough, it could provide a relevant technology for implementing analog and digital circuits as well as neural networks.

References

1. Shibata, T., Ohmi, T. An Intelligent MOS Transistor Featuring Gate-Level Weighted Sum and Threshold Operations Technical Digest, International Electron Devices Meeting. (1991)
2. Hasler, P., Lande, T. S. *Overview of Floating-Gate Devices, Circuits, and Systems* IEEE Transactions on Circuits and Systems II: analog and digital signal processing, special issue on floating-gate devices, circuits, and systems. (2001)
3. Austria Mikro Systeme International AG *0.6 um CMOS CUP Process Parameters* Document No. 9933011, Rev. B, October 1998.
4. Berg, Y., Wisland, D. T., Lande, T.S.: Ultra Low-voltage/Low-Power Digital Floating-Gate Circuits. IEEE Transactions on Circuits and Systems II. (1999)
5. Aunet, S., Berg, Y., Sæther, T. Floating-Gate Low-Voltage/Low-Power Linear Threshold Element for Neural Computation. Proceedings of the 2001 IEEE International Symposium on Circuits and Systems. (2001) 528-531
6. Aunet, S. Real-time reconfigurable devices implemented in UV-light programmable floating-gate CMOS Dissertation for the degree of doktor ingeniør, Norwegian University of Science and Technology. (2002) ISBN 82-471-5447-1.
7. Berg, Y., Lande, T. S., Næss, S. Low-Voltage Floating-Gate Current Mirrors Proceedings of the the Tenth Annual IEEE International ASIC Conference and Exhibit. (1997)
8. Andreou, A. G., Boahen, K. A., Pouliquen, P. O., Pavasovic, A., Jenkins, R. E. Current-Mode Subthreshold MOS Circuits for Analog VLSI Neural Systems. IEEE Transactions on Neural Networks. (1991)
9. Aunet, S., Berg, Y., Tjore, O., Næss, Ø., Sæther, T. Four-MOSFET Floating-Gate UV-Programmable Elements for Multifunction Binary Logic Proceedings of the 5th World Multiconference on Systemics, Cybernetics and Informatics. (2001)
10. Shams, A. M., Bayoumi, M. Performance Evaluation of 1-bit CMOS Adder Cells. Proceedings of the 1999 IEEE International Symposium on Circuits and Systems. (1999)
11. Shams, A., Bayoumi, M. A Framework for Fair Performance Evaluation of 1-bit CMOS Adder Cells. Proceedings of the 42nd Midwest Symposium on Circuits and Systems. (2000)
12. Rabaey, J., Pedram, M., Landman, P. Low Power Design Methodologies. in J. Rabaey, M. Pedram (editors), Low Power Design Methodologies, Kluwer Academic Publishers. (1997).
13. Soeleman, H., Roy, K., Paul, B. C. Robust Subthreshold Logic for Ultra-Low Power Operation. IEEE Transactions on Very Large Scale Integration Systems. (2001)
14. Beiu, V., Makaruk, H. E. Deeper sparser nets can be optimal Neural Proc. Lett.(1998)
15. Beiu, V., Quintana, J. M., Avedillo, M. J.: VLSI Implementations of Threshold Logic: A Comprehensive Survey. to appear in IEEE Transactions on Neural Networks - Special Issue on Hardware Implementations, to be published

CMOS Implementation of Generalized Threshold Functions

Marius Padure , Sorin Cotofana[2], and Stamatis Vassiliadis[2]

"Politehnica" University of Bucharest, Romania
Delft University of Technology, The Netherlands

Abstract. Threshold Logic (TL) gates can evaluate any linearly separable function via the computation of a weighted sum over the input variables. In this paper we generalize this mechanism and introduce the novel concept of -order Generalized Threshold Logic (GTL) gates. Such a GTL gate has augmented computational capabilities as it can evaluate a weighted sum of -term AND products over the input variables. Additionally, we propose an implementation scheme for second-order GTL gates in CMOS technology. To assess the practical implications of the augmented computational capabilities of GTL gates we present a one gate implementation of -input parity function and a scheme to compute the block carry-out function utilized in carry lookahead addition algorithms. Our results indicate that the -order GTL gate based implementation of the carry-out for a -bit block requires transistors in each data and threshold mapping bank as opposed to transistors required by a standard TL gate based implementation.

1 Introduction

A Threshold Logic Gate (TLG) is defined as an -input processing element such that its output performs the following Boolean function[3]:

$$f \tag{1}$$

$$f \quad n \quad T \tag{2}$$

where n, W n and T are the set of Boolean input variables, the set of fixed signed integer weights associated with data inputs, and the fixed signed integer threshold, respectively [6]. The vector formed by the group of weights , , n and the threshold T, whose absolute values are the minimum, is called the *basic structure* of the threshold function and is presented as follows: n n $\mathbf{T}$.

Speaking from the device/circuitry point of view the main advantage of TL gates when compared to standard Boolean gates is the parallel processing due to internal

[3] All the operators are algebraic. In order to avoid confusion throughout the text, the Boolean And, Or, and Xor operators have designated the symbols, $\wedge$, $\vee$, and respectively.

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 65-72, 2003.

multiple-valued computation of the weighted sum. However when compared to classical multiple-valued logic circuits TL gates are more robust against parameter variation because the TL gates inputs and output are still digitally encoded. Given that TL gates are fundamentally more powerful than the conventional Boolean gates and that there is evidence of direct implementation of Threshold Logic gates in CMOS technologies [1, 3, 5, 7], TL based design of computer arithmetic related Boolean functions has been the subject of numerous studies [2, 4, 10]. Furthermore, recent investigations [8], suggested that a hybrid Boolean and Threshold Logic design approach can be utilized for the design of certain computer arithmetic building blocks, such as counters and compressors, and provides a substantial speedup when compared with pure Boolean and pure Threshold logic implementations.

In this paper we continue our investigation on hybrid Boolean and Threshold logic schemes and introduce a Boolean logic augmentation of the computing capabilities of the CMOS threshold logic gate presented in [7]. That is to say we apply the hybrid paradigm at the gate level rather than at the network level. The gate we propose embeds a mechanism to evaluate a number of Boolean AND products over its inputs. Thus the differential latch based topology can evaluate a weighted sum of AND products over the inputs instead of a weighted sum of inputs, i.e., it can evaluate *Generalized Threshold Logic Functions*. The extra computation step allows for the construction of faster threshold logic networks since the higher computational power embedded inside the gate compensates the gate latency degradation. The main contributions of the paper can be summarized as follows:

- We introduce the general concept of -order Generalized Threshold Logic Gates.
- We propose a CMOS implementation of second-order Generalized Threshold Logic Gates (GTLG^2).
- We discuss the implications of the GTLG^2 design style when utilized for the implementations of computer arithmetic relevant functions, e.g., XOR, carry lookahead computation.

The paper is organized as follows: Section 2 introduced the definition of the -order Generalized Threshold Logic Gates and propose a CMOS based implementation of second-order Generalized Threshold Logic Gates (GTLG^2). The GTLG^2 basic operation and its underlying principle are explained in detail. Section 3 discusses the implications of the augmented computational capabilities featured by the GTLG^2 gates and presents as a case study the implementation of a block carry lookahead algorithm with TL gates and GTL^2 gates. Section 4 presents some concluding remarks.

2 CMOS Generalized Threshold Logic Gates

As suggested in Equation (2) a traditional TL gate can evaluate any linearly separable function via the computation of a weighted sum over the input variables. We propose to generalize this concept by replacing the input variables with AND products over the input variables. This new class of functions constitutes a superset of the class of linearly separable functions and the evaluation of such generalized functions requires the utilization of generalized threshold gates operating according to the following definition:

Definition 1: A -order *Generalized Threshold Logic Gate* (GTLG) is a device which performs the following Boolean function:

$$\left\{ \right.^{n} \qquad T \tag{3}$$

where $_{n}$ **T** is called the -order GTL gate basic structure. The GTL gate inputs are $\{$ $_{n}$ with each , being a -term Boolean input set, $\{$.

Any -term set has an associated weight value but the effective fan-in of the gate can assume values between and as not all the should be different. Moreover, although not suggested by the Definition 1, a GTL gate can accommodate input sets with literals which implies that inputs from the corresponding -term set should be mapped to logic one. It can be easily noticed that Threshold Logic functions are a subclass[4] of GTL functions since results in $\{$, then the product in Equation (3) becomes and Equation (2) is obtained. Therefore, it is expected from GTL gates to perform a broader range of Boolean functions overlooked by the previous investigations carried in Threshold Logic functions.

In the remainder of the section we discuss the CMOS implementation of such GTL gates. We restrict the presentation, without any loss of generality, to the CMOS implementation of second-order Generalized Threshold Logic gates (GTLG^2). However, -order GTL gates can be easily constructed by extending the circuit principles presented in the following subsections. As the scheme we propose is based on a Differential Current-Switch Threshold Logic (DCSTL) gate circuit, we first present the DCSTL. Second, its extension to a second-order GTL gate is presented and its operating principle is explained in detail.

2.1 Differential Current-Switch Threshold gate

A schematic diagram of the Differential Current-Switch Threshold Logic gate (DCSTL), [7], is presented in Fig. 1. The circuit comprises a fast semi-dynamic CMOS latch and two analog computation blocks referred as *data bank* and *threshold mapping bank*. Both data and threshold mapping bank consist of two parallel-connected sets of unit nMOS transistors ($_{\ldots n}$ and $_{\ldots n}$).

The main principle of such gate stands in the fact that the transistors from both banks are operated in the linear region. Therefore their drain current depends mainly on the multiplicity factor, , and both transistor's overdrive, $_{,n}$, and drain-source voltage, , [11]. Assuming that, in normal operating conditions, the nMOS transistors from both banks have equal $_{,n}$ and imply that each nMOS transistor generates a current proportional with its multiplicity factor. The total currents generated by the transistor banks ($_{d}$ and) are compared each other by the latched comparator and therefore the node is logic zero if the current generated by the data

[4] To comply with already existing definitions and to avoid confusions, we use in text the more common term "Threshold Logic functions", although "GTL functions" would be more appropriate in the context of our presentation, according to previous definition.

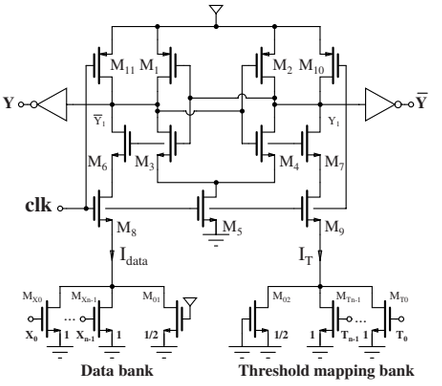


Fig. 1. CMOS Differential Current-Switch Threshold Logic gate

bank, $_d$, is greater than the current generated by the threshold mapping bank, , and logic one otherwise. We note here that the data bank is prevented to have similar current with the threshold mapping bank, when the threshold is reached, since the nMOS transistor having the multiplicity is always on. The DCSTL gate is operated as follows. On the falling edge of the clock, the flip-flop enters in *precharge phase*. Therefore, are on, nodes and are precharged high, and the outputs and are both zero. On the rising edge of the clock, the flip-flop enters the *evaluation phase*. Therefore, , are on and (shutoff devices) start drawing currents from nodes and . If $_d$ then the voltage at node will start to drop faster than the voltage at node . Therefore, crosses first the latch switching threshold which regenerates rapidly to low and high, causing high. Conversely, if $_d$ then low and high, causing low. At the end of the evaluation phase, the high-rising node among and will be electrically decoupled from being connected to ground by one of the shutoff transistors going off. Therefore no DC power is dissipated at the end of the evaluation phase.

The gate is *run-time re-programmable*, since the Threshold Logic function can be changed dynamically between two evaluations, *differential*, and it is characterized by the fact that the implementation of threshold is particularly similar with the weighted sum as T ^{n} This last feature is particulary useful when implementing negative weights. For example, a DCSTL having an input with a negative weight, , can be easily derived from a DCSTL gate with an identical absolute weight value, , by simply connecting the corresponding data input into the threshold mapping bank. Of course, the capacitive balancing condition (the total sum of transistor widths must be identical in both nMOS banks) must hold true and consequently dummy transistors must be added in the data bank.

2.2 Second-order Generalized Threshold gate

The second-order Generalized Threshold Logic gate can be derived from a DCSTL gate if the data bank and the threshold mapping bank are replaced with the banks depicted in Figure 2.

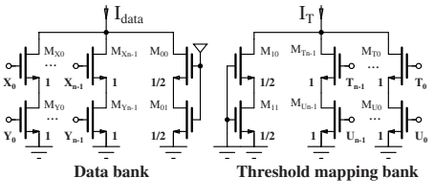


Fig. 2. GTL CMOS gate - data and threshold mapping bank

Very similar with a DCSTL gate, each analog processing block consists of a set of parallel-connected nMOS transistors. In contrast with DCSTL gate, each nMOS transistor from both banks is actually a series of *two* transistors (,⋯n , ,⋯n), and (,⋯n , ,⋯n). Therefore, since each series of two nMOS transistors implements actually an AND function of two inputs, GTL^2 gate allows a more general computation in both banks, i.e., the weighting operation, specific to a simple TL function, is no longer performed on the input basis but on AND products containing at most literals. The GTL^2 structure can be easily extended to GTL gates, case in which the two banks are formed from branches of transistors[5] in series.

The GTL^2 gate from Fig. 2 has in general data inputs, ⋯n , ⋯n , and threshold mapping inputs T ⋯n , ⋯n . Since two nMOS transistors in series are electrically equivalent with a single transistor with a smaller driving current, [11], it is expected a latency degradation of a GTL^2 gate when compared with a TL gate. However, when such GTL gates are employed in bigger circuits, the extra computation step, i.e., AND products of terms, is expected to actually *increase* the speed of those circuits since the higher computational power embedded inside the gate compensates the latency degradation as the width of the AND products increases.

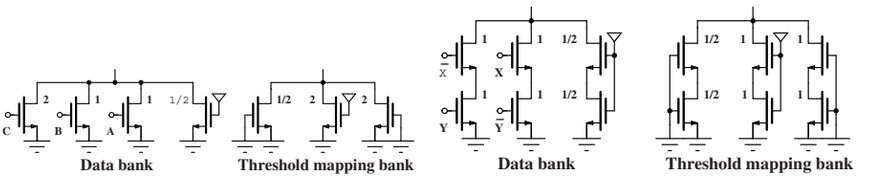
3 Discussion

In this section we evaluate the potential implications of the GTL^2 gate on the implementation of some computer arithmetic relevant functions. First we discuss how the generalized gate can be utilized to extend the class of the Boolean functions implementable in one TL gate with the parity function. Subsequently, we analyze the utilization of GTL gates in carry lookahead adders and evaluate the potential area reduction this approach may bring when compared with a TL gate based one.

3.1 Boolean functions implementable in a single GTL gate

While simple AndOr, OrAnd functions (and their inverted versions) can be implemented in a single TL gate (see Fig. 3(a) for a schematic of data and threshold mapping banks) as they belong to the class of linearly separable functions, this doesn't apply for

[5] Although the GTL^k gates have, by definition, the inputs sets of equal width they can implement actually a weighted sum of variable-length AND products over the inputs. Of course, if an AND product has the length , the other inputs from the same transistor branch should be *mapped* to logic one all the time.



(a) (b)

Fig. 3. Various TL and GTL functions implementations: a) ∧ ∨ ; b)

the implementation of other functions, e.g., (inverted-)parity, one of the most important Boolean function for computer arithmetic. However, due to its augmented computational power one GTL gate can implement such symmetric functions. Therefore TL circuits employing GTL gates are expected to be *faster* due to the reduced number of gate levels. As an example we present in Fig. 3(b) the data and threshold mapping banks for a GTL^2 gate configured to perform the XOR function of two Boolean variables. The implementation in Fig. 3(b) assumes that the inputs (,) are in dual-rail form which usually implies *wiring overhead* at circuit level. We would like to mention that dual rail is not mandatory for such an approach as a negated variable in a Boolean function can be replaced in general in TL by its complement using the relation $\overline{A}$ A. This property together with the fact that the threshold mapping bank can also accommodate input signals allow for one GTL^2 gate implementation of a AndOr22 function having a negated Boolean variable without requiring a dual rail design style. To exemplify this approach we describe in Eq. (4) the GTL^2 implementation of an AndOr22 function, having a negated Boolean variable.

$$A \quad \overline{\quad} \quad D \quad \{A \quad D \quad A \tag{4}$$

3.2 Carry Lookahead Addition

Let us assume we want to implement an -bit binary adder using the Carry Lookahead Addition algorithm [9]. Considering that operands and are divided in groups of bits each, the carry output signal of the group can be computed as follows:

$$c \quad {}_2 \quad {}_2 \quad {}_2 \quad c \tag{5}$$

where and , $\overline{\quad\quad}$ are the *generate* and *transmit* signals of each pair of bits, , respectively, and c being the carry output from the previous -bit group. The logic expression in Eq. (5) implies is that a carry is generated out of each -bit block if a carry is generated on the last position ( ), *or* a carry is generated on position ($_2$) *and* it is transmitted through the position ( ), ..., *or* a carry is generated on the first bit position ( ) and it is transmitted through all the next bit positions (, ,), *or* a carry input is present (c) *and* it is transmitted through all the bit positions

of the block. It can easily be observed that this logic expression contains products of at most logic variables. Moreover, this logic expression can be evaluated with a Threshold gate with its basic structure defined by the following theorem.

Theorem 1: A simple TL gate which implements Eq. (5) and having the inputs $_2$ $_2$ c has the basic structure:

$$_2 \quad _2 \tag{6}$$

Proof: By induction.

basis: Trivial with simple substitution. Since c c , it is easily to prove that is implemented with a TL gate with the inputs c **T** and the basic structure .

step: Assume that the theorem holds true for prove that it is also true for . Assuming that the theorem holds true for , it is implied that a TL gate with , ^{2}, ... and T computes c . Given the carry recurrence formula: c c it follows that and (i.e. term dominates while and c terms have the same weight). In considering c , its dominant term ( ) has the weight , identical with . This implies and .

>From Theorem 1 it can be easily concluded that the computation of the carry-out of a -bit block requires a DCSTL gate with transistors in each data and threshold mapping bank. In the following theorem we analyze the GTL gate based computation of the carry-out in Eq. (5).

Theorem 2: The GTL gate which implements Eq. (5) and having the inputs:

$$_2 \quad _2 \quad _2 \quad c \tag{7}$$

has the following basic structure:

$$\mathbf{-} \quad \mathbf{-} \tag{8}$$

Proof: The Boolean Eq. (5) is expressed as a sum-of-products form with products, each term having at most literals. The Boolean function is logic one if *at least one* term is logic one. Therefore a GTL gate configured with a threshold of and having weights of associated with each AND product can evaluate Eq. (5). The assignation of the GTL gate inputs is as follows:

$$\begin{array}{cccc} 1 & _2 & _2 & _2 \\ 1 & 1 & & \\ 1 & 1 & 1 & \vdots \\ 1 & 1 & 1 & \\ 1 & 1 & 1 & c \end{array} \tag{9}$$

where by it is designated a logic one input.

>From Theorem 2 we conclude that the computation of the carry out of a -bit block can be implemented with a GTL gate having ^{2} transistors in each data and threshold mapping bank since GTL gate requires for each AND product transistors and there are products. This constitute a substantial area reduction when compared with transistors required by an implementation based on a standard TL gate.

4 Conclusions

In this paper we introduced the novel concept of -order Generalized Threshold Logic gates and presented an implementation of second-order Generalized Threshold Logic gates in CMOS technology. To assess the practical implications of the augmented computational capabilities of the new GTL gates we presented a one gate implementation of -input parity function and a scheme to compute the block carry-out function utilized in carry lookahead addition algorithms. Our results indicated that the -order GTL gate based implementation of the carry-out for a -bit block requires ^{2} transistors in each data and threshold mapping bank as opposed to transistors required by a standard TL gate implementation.

References

1. M.J. Avedillo, J.M. Quintana, A. Rueda, and E. Jimenez. Low-power CMOS Threshold-logic gate. *Electronics Letters*, 31(25):2157–2159, December 1995.
2. V. Beiu, and J.G. Taylor. On the circuit complexity of sigmoid feedforward neural networks. *Neural Networks*, 9(7):1155–1171, 1996.
3. P. Celinski, J.F. López, S. Al-Sarawi, and D. Abbott. Low power, High speed charge recycling Threshold Logic gate. *IEE Electronics Letters*, 37(17):1067–1069, August 2001.
4. S. Cotofana and S. Vassiliadis. Periodic symmetric functions, serial addition and multiplication with neural networks. *IEEE Trans. on Neural Networks*, 9(6):1118–1128, October 1998.
5. Y. Leblebici, H. Ozdemir, A. Kepkep, and U. Cilingiroglu. A compact high-speed (31,5) parallel counter circuit based on capacitive Threshold-logic gates. *IEEE Journal of Solid-State Circuits*, 31(8):1177–1183, August 1996.
6. W.S. McCulloch, and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.
7. M. Padure, S. Cotofana, C. Dan, S. Vassiliadis, and M. Bodea. A low-power Threshold logic family. *IEEE International Conference on Electronics, Circuits and Systems, ICECS 2002*, 2:657–660, September 2002.
8. M. Padure, S. Cotofana, and S. Vassiliadis. High-speed hybrid Threshold-Boolean counters and compressors. ^{t} *IEEE Midwest Symposium on Circuits and Systems*, in press, 2002,
9. B. Parhami. Computer Arithmetic - Algorithms and Hardware Designs. *Oxford University Press*, 91–107, 2000.
10. S. Vassiliadis, S. Cotofana, and K. Bertels. 2-1 addition and related operations with Threshold logic. *IEEE Transactions on Computers*, 45(9):1062–1068, September 1996.
11. N. Weste and K. Eshragian. *Principles of CMOS VLSI Design*. Addison-Wesley Longman, 1993.

A A A 4 g peed p y d y e d g Adde

t l s ' S t d tt[2]

n r or g r or anc In gra c no og S c
n r or o ca ng n r ng
D ar n o c r ca an c ron c ng n r ng
n r o a S 00 ra a
e i i e e e g e i e e u u
o r ng n r ng Gro c r ca ng n r ng D ar n
D n r o c no og w g D D N r an
i e et tu e t

A st t g na c a r a D r
o og c r D r n a r a on a r
carr oo a a /carr c c ng r o og c an con n
ona S og c D wa gn an a n a 0
roc wor ca cr ca a a nc 0 w c own o
on a rag 0% a r an r o ro o g oo an
na c og c a r w a a r c ng ran or
co n on a rag o r 0% co ar o a a r

n uc i n

s ld l w s t d d d d s d t s
s p s d t s d d l d pt d t t p d t
t l l t s d d s ffi t p s l l z t s s
t t t s t s d l ttl p t S ffi t t
l z t s tl l l d s ll ppl t s
s d t s d st t d ts l t t p t sp d
d s tl d d dd t ll t w s s st d t t
d d t lst t S l d t d l p d
t p p st t S pp t s p ll l p p l t
t s w l s l t t ld l s p ss s
d s d s dd ss t s ss p p s 6 t dd
dd t s t st t l p t s p d S p s
s s s p p p p s s 6 t dd s tw ld d s
l d s l t d d l s ld
d d S l s t d s t st st p ss l d
6 t dd t t t t d t s d l t t s
p p s d 6 t dd s s l t d t l p t d l 6 ps
p ss

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 73-80, 2003.

S t w t s ld l s s
ll w d d s pt S t S t d l ps t p p s d
d l d s l t s s d S 6
t dd t d S t s lts t p p s d 6 t dd t s
p s t d ll l s s S t 6

s ic

t s ld l t s t ll s l t d l t t
t s p ts $_2$ $_n$ d p d s s l tp t
l w t d s t p ts s p t d ll w d t s ld
p t
l t p t d s t s ll d t s ld t
d t s sp d t t t s ld T d t w ts $_2$ $_n$ w
s t w t sp d t t $_t$ p t l tp t
s

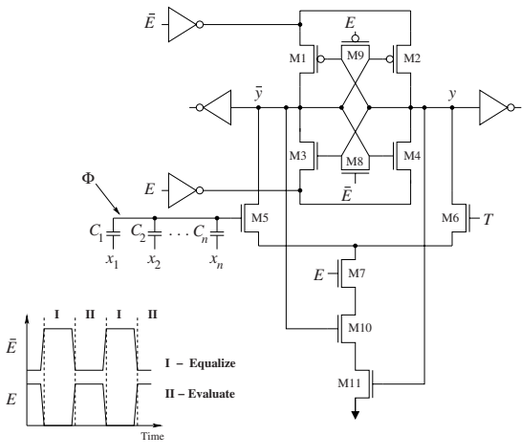
$_n$ T

t w s

t p d t l z d st t l t s
d st t t s ld T pl p t t w t t w ts
d T w ll l z p t t d s tt T t
t p t s t t s s t l t s t t ff s s
tl s d p t t l p l t t l
l S tl d d d s d t sp d t p t
t ll t d sp ll ppl t s l p t
l s s s p t t t s s d t d p
t l s lts 6 w s st d t s t l l
l

C cyc in s ic

l z t S t s ld t s p s t d d s d t p
p s d dd s w d s d s ws t t st t s s
pl ss pl d t s st s t s tp t d ts pl
t d l t s sp d t l l s l d
ts pl t p ts p t l pl d t t t t
d t t s ld s s t t t lt T 6 p t t l
s $_n$ $_o$ w $_o$ s t s ll p t s
l d p s t s t t t d t l s t s l z d s t
t p t s t pp p t l s p ll S t l t s
p t s pl t d tw t p l s l d p l s l l s
l s l t ls t p d t t t s s
t t s tw p s s p t t l z p s d t l t
p s s t tp t lt s l z d s t



F R ga c rc an na gna

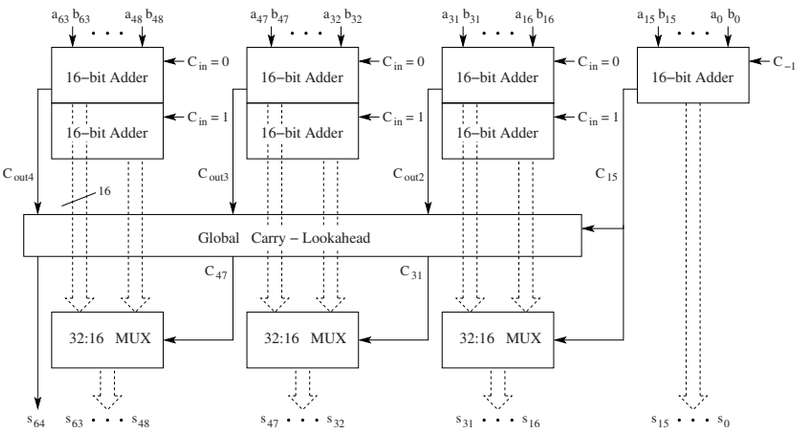
tp ts d s t d d t d ff t l t d ws d ff t ts t l l z d d s d s s pl s t t d t t d l t l t s d pl s t d ff p t t l w p s t tw d l t t t s t t s w t t st t d t s w t t w t d s t p ts s t l ss t t t s ld T d t s l z d s st s d t ff t d ff t l t t l t s pl t d t d t p w d ss p t t w s s w t l l p t t sp d

i A i i n sin C yn ic CMO

d l pp t p p s d 6 t dd s t t l z t t t l l d s d t l S l d l s t p p s pt z d d s 6 t dd s d d S pp

t t t t l l t d s s st t d t pt z sp d s d t d t s d l t t s t l t s s d st t d d S d t t p t d p t t l d d s l t s s d t t pl t t l l w t t l s p s t 6 t dd s s d t t t l p t d l d w p ss l st t d d S s d t d t l d t p p s d 6 t dd s d p t d t s s t 6 t p t dd ds d d d t 6 t dd l s 6 t l t s s l s s ls t p ts t l l l d t w t s t s l t s ls c c_4 d c_4 s s l t s ls s d t s l t t pp p t 6 t l s s s t 6 lt pl s t l p t t 6 t dd s sts t

t t tp ts 6 t l t l l l d
t w s l ts t st s t 6 t t p t d s d 6
U d l
t t p t d s s p t d t 6 t l s s p s
6 t dd s w t c_n d c_n 6 t t p t d s s st
d t l l s t p t s l t t s t s l
l p t t s t t 6 t dd s d t l l s
s d t d d l w d t t s p t t
t 6 t p ts t 6 t w d l s



F 2 D oc agra

t l w st t l p t d l t p t t t tp t
6 t dd d t l l s st s st s p ss l t
t s p ss l t p t t s l s 6 t l t
s t s st p s t d w t s l ds t t
t s w ts t t w s p t l l
t st t t t t s s t 6 t p t t s
p d tw l ls t s s d s ss d t ll w s t

e he er

d s t 6 t dd s s s l d s s w ll w t
d s l t d pt dd t tw s d s t
sp d 6 t dd t p l t s t l z d t pl
t l d p lls w p t p t d p
p p t s ls l p t ts
d s t p lls s d t s
w ts s s l t d t l l p s w t w t
p t w ts pl t d d s tl t s l t t

s s pl d t p ss s t d s
s w ts s ts t lt w s d t s l d
t s s pl t t t s s w l t t s
w ts t pp t l d s t s s t ss w t pt z t
t l p t d l t 6 t dd s s w ll s w l ds t t
p lls t w d p ts
p t t s *c* p ss d p t t t s
p t d p p p t s ls t t p s t s
ll ws

w d t p t dd d ts *c* d t s t
t d t p s t d *c* d t s t t t S p s t ss
t t *c* t t s l t p s t *c* s *c*
d t l l pp t pl t t s d
pl st t S s t p t l s l w d l t
6 s t t t d t ss l t t S
s p t t t s t p t p ds
p d t t l s st s t p t d p
p p t s ls t p d p t d d tl
t p t p ds t l z s t t s s t l
pl t t t s ld l s t d d s t d
p t s t tw t w ds t p ds s st tl
t t t p s t l s ts [4]
S l l t p p p t s t t p
l w s t s t tw t w ds t p s l t
t t s w tt t s

s ⎧ 6
s ⎩

p t w ts t t s w l l t d t s
d t t t s lds d ff s s t t t p t w ts
s d tw tw t s t d t s d t
s w ts t w ts t p sl st t d t
pl t p t d p p p t s ls
p t d t t p t p t dd d ts s ll ws

s { 2 2 6 6

s { 2 2

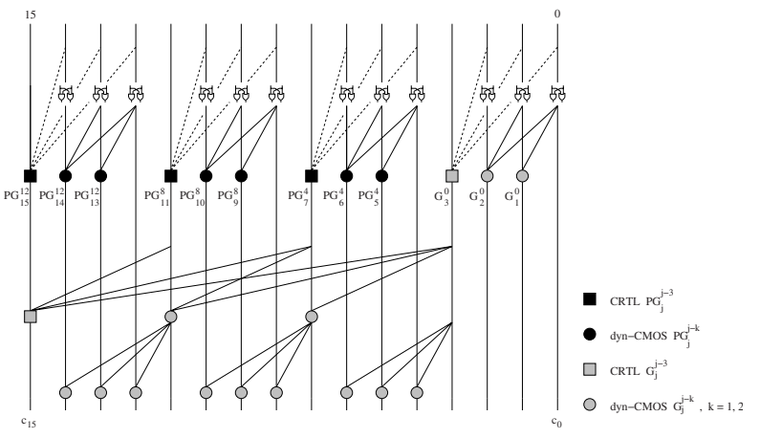
t l st s t p t d p p p t s ls
t p p t d s t d S pl t t
t s s d s ls p t d st t S
t s d l l s l t ss l t s t t p t
d p p p t s ls t p t t st s t *c* s l
l ws

2 2 2 4 2 4
s 2 2 4 4

s ts p t d s st t S t s

c

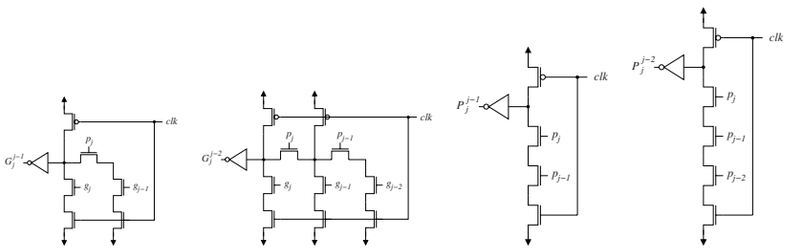
p t t t l s s p d p ll l w t t l l t
t s 6 t l s ts l l t d p ll l
w t t 6 t dd l l l d



F a r carr r x r c a c

s t d t p p s d 6 t dd p t s
s w t l p t s s p lls t p t t
p p t d t s ls ps ts s
t s d s d t d t l s s l s
ll s sts tw t s w t t s lds d ff d s t
s p t tw p t t w t d s p ts s s
p s t p t lls S tp t s p t d s
s l p t ll lls t p t

w t t t l p t s st t l d S
t d p p t lls t d ll s d t d
l s d t ts t s lls



a c

F D na c S c rc or a $\frac{j}{j}$ $\frac{j}{j}$ c P_j^j an P_j^j

su s f p s i A si n

t l p t t 6 t dd s sts tw t s t p t c
d S t t p t c_4 t l l l d l d
t p d U s d t s l t t s ts $_4$ t l p t
l t w s s l t d s p ss p t s S
d w s d t pp t l 6 ps s t l l t d
t t p p s d dd s t s t w p
w t t d l t tl p t d dd s t d l w s l z d w t
sp t t t t t d l d l s t d l
s z d t d s z d t s d t
l z d d l t l l s l t l st t w d
p ss s U d t p l t l d t s d l t
p ss s d t s w s pp t l 6 ps st t t
st t t t l t s st s t 6 t dd w s ls l l t d

o ar on o D w o r g a r

	Naff g r 0	oo *a*	S n *a*	D
ran or		0	0	
Nor a In D a				

t s st t d l z d d l p s w t t t
d sp d dd s s s w l t s ld t d t t t d l
s lts p s t d S *l* s d s l t d t s lts t
ffz d *l* p s ts s lts
l s st sp d p l st % d d t
t t s st t % p d t t sp d dd d s s

C nc usi ns

sp d 6 t dd s d d l d s l t s
s l s ld d d S s p p s d
w st s t l p t d l d t s st t w s w t t
p d pp t l % p d t p sl p p s d
t l d l sp d dd s s lts s w t t
d t l S l w t t pp p t t t l st t
l t l st d p t t t ts d

f nc s

r a r c n rog ca ac
r o og c ga I JSS
n o J Saraw S o D ow ow r g c arg
r c c ng S r o og c ga I c ron c r 00 0 0
a r o o ana S a a S ow ow r r o og c a In
roc I In rna ona on r nc on c ron c rc an S 00
0
a r o o ana S a a S g r r o oo an
og c co n r an co r or In roc ng o I In rna ona
w S o on rc an S 00 0
c r n rog co ac g
ara co n r c rc a on ca ac r o og c ga I JSS

n o J Saraw S o D ow carr oo a a
a on ng c arg r c c ng r o og c In roc I In rna ona S
o on rc an S o n x 00
a a S o o ana S r a on an r a ar c o r
a on w r o og c I ran o r **5** 0 0
Ra a J D g a In gra rc D gn r c r n c a
N w J r S
o R a orow r o w r roc ng o I
00 0 0
0 Naff g r S n nano con 0 a r gn In In rna ona So
S a rc on r nc D g o c n ca a r I
oo R S J oo J 0 na c ow ow r a r gn In
roc ng o In rna ona S o on rc an S Sw r an
I 000 I I
S n ao a g 0 S a r gn
In roc ng o SI I 00

Validation of a Cortical Electrode Model for Neuroprosthetics purposes

García-de-Quirós F.J [1], Bonomini, M.P.[1], Ferrández, J.M.[1,2], Fernández, E. [1]

[1]Instituto de Bioingeniería, U. Miguel Hernández, Alicante,
[2]Dept. Electrónica y Tecnología de Computadores, Univ. Politécnica de Cartagena.

Corresponding Author: fgarciani@nexo.es

Abstract. A study of the electrical parameters of a microelectrode array designed for chronic current stimulation through cortical tissue was carried out. The microelectrode array was inmersed in ringer solution for measurements. Resistive and capacitive components, as well as total equivalent impedances were analyzed at different current levels using Electrochemical Impedance Spectroscopy (EIS) techniques. A SPICE equivalent model was introduced. Our results show that experimental registered data matches the proposed Warburg model for both low and high frequencies denoting how electrode impedances are affected with frequency variations. Moreover, effects produced by injecting different amount of charge were analysed.

1. Introduction

The possibility of interfacing the nervous system with electronic devices has long fascinated scientists, engineers and physicians. An electronic device which restores a motor or sensorial function is an important research field for bioengineers nowadays. Present design and manufacturing of 3D electrodes may be considered as a real choice for neuroprosthetic devices intended to the recovery of motor and sensory abilities. However, this approach must fulfil several conditions:

1) Tissue must accept the device (*biocompatibility*).
2) Electrical stimulation of nerve tissue must be induced by means of a charge displacement (*polarization*), which is performed efficiently through current stimulation.
3) Current injection in neural tissue must be effective and safe.

Our main goal is to analyse the electrical behaviour of a penetrating microelectrode array with a three-dimensional architecture (Utah Electrode Array-UEA). This microelectrode array has been used extensively in acute and chronic recording experiments, but few information is known about its characteristics regarding working frequency, current and voltage spans, etc.

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 81-88, 2003.

2. Electrode-Saline Interface Modelling.

The electrode-medium interface has been over the past years extensively studied in electrochemistry to characterize electrodes in applications such as batteries, reactors, and so on. Geddes and Warburg determined the nature of the impedance model for these electrode-electrolyte systems, based on the different chemical processes taking place for many different combinations of voltages and frequencies. In this case, the classic Randle's equivalent circuit (Figure 1) seems to be appropriated.

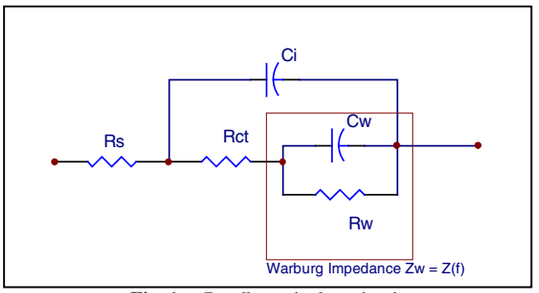


Fig. 1. – Randle equivalent circuit.

This equivalent circuit comprises a capacitance C_I in parallel with a resistance R_{ct} and the Warburg difussion impedance Z_w. The R_s resistor represents the resistance of the electrolyte solution. When a metal and electrolyte come into contact, a charge region is formed in the electrolyte at the interface. Initial theories developed by Helmholtz assumed the charge of solvated ions to be confined within a rigid sheet next to the metal surface, being this charge equal and opposite to that in the metal. This capacitance is determined by:

$$C_H = \frac{\xi_0 \xi_r}{d_{OHP}}$$

Where C_H is the capacitance per unit area (F/m^2), ξ_0 is the free space permitivity ($8.85 \cdot 10^{-12}$ F/m) , ξr is the relative permitivity of electrolyte and d_{OHP} is the Helmholtz plane distance. It is settled that for physiological saline at 25ºC, d_{OHP} average value is 5Å, and ξr value is in [6,7,8] average range, yielding a worst case capacitance of 0.14 $pF/\mu m^2$. Therefore, for an electrode area of $3{,}1 \cdot 10^{-4}$ cm^2, we get a C_H value of 4.34 nF.

On the other hand, if a DC potential is applied across the interface, a current may flow under certain conditions. The flow of such current across the interface implies net movement of charge in response to the electric field. This net charge transfer is modelled by R_{ct}, and represents no-faradaic processes in the metal-electrolyte interface. This resistance depends on the current density (J) across the interface, the thermal voltage (V_t) of charge distribution and ionic valence (z) of predominant species. This is determined by:

$$R_T = \frac{V_t}{J_0 z} = \frac{\eta_t}{J} \quad (\Omega \cdot cm^2), \quad J = J_0 e^{(z\eta_t / 2V_t)} (A/m^2)$$

The value of R_t when little stimulation voltages are applied (about mV) is well modelled by the first term over all the overpotential (η) and frequency range. Taking $J_0 = 7{,}94 \cdot 10^{-4}$ A/cm^2, z = 1, Vt = KT/q, and an electrode area of $3.1 \cdot 10^{-4}$ cm2 , we get a R_t of 106,451kΩ.

The Spreading Resistance Rs models the effect of the spreading of current from the localized electrode to a distant Ag/AgCl reference electrode in the solution. Rs is given by $R\ (\Omega) = \rho \cdot L/A$, where ρ is the electrolyte resistivity in Ω·cm (72 for physiological saline at 25ºC), *L* is the length (cm) and *A* is the cross surface (cm^2) of the solution through which current passes. Several calculations were made over several electrode topologies, nevertheless, using the approximation of a circular electrode calculated by Newmann (9) $R\ (\Omega) = \rho \cdot (\pi)^{1/2} / 4(A)^{1/2}$ we get values lower than 1KΩ. To get an empirical approximation of R_s, we use a potentiostatic method (Figure 2) consisting in the application of a square voltage waveform to the electrode in ringer solution. In the initial transients, we can appreciate the interface capacitor charging as a current spike, afterwards the current comes to nearly zero due to a high R_t value. Several measurements were taken on different electrodes with several voltage and frequency values. A good average value is 500Ω.

The Warburg impedance Z_w shows a frequency dependent behaviour in an inverse square root proportional factor:

$$Z_w = K \frac{1}{(j\omega)^{1/2}} = \sigma_1 \omega^{-\frac{1}{2}} - j\sigma_2 \omega^{-\frac{1}{2}}$$

σ1,σ2 Resistive and capacitive Warburg coefficients

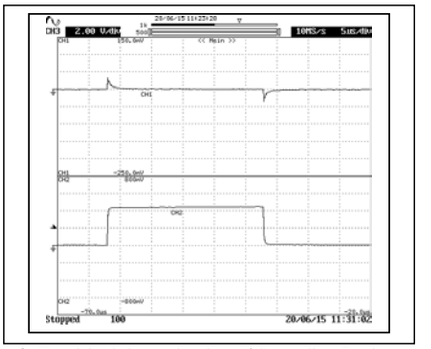


Fig. 2. Empirical determination of Spreading Resistance (R_S).

The Warburg impedance models the diffusion effects which play an important role when low frequencies and important amplitudes are chosen for voltage stimulation. This produces charged ions far from the ionic cloud to arrive to the electrode

neighbourhood, participating in interface impedance. After a straightforward algebra we get:

$$Z_W = \frac{1}{R_W} + j2\pi C_W \Big)^{-1} = \sqrt{2} R_{W\ \angle -45^\circ} \quad , \quad R_W = \frac{10^3 V_t}{z^2 qn^0 \sqrt{\pi f D}} \left(\Omega \cdot cm^2\right)$$

Where f is frequency (Hz), D is the diffusion coef. (cm^2/s) of the predominant ionic species and Z is its valence. Therefore, Z_w depends on frequency and ion concentrations. Its characteristic behaviour is a 45º slope ruling over the Nyquist impedance diagram at low frequencies. For an electrode area of $3.1 \cdot 10^{-4}$ cm2 we get a value about $22.68_{\ -45^\circ}\ \Omega$ (for f < 1 kHz).

The following figures show the Nyquist plots for low and high impedances from the proposed model, it clearly demonstrates that the circular plot is produced by the predominant role of R_{CT}, R_S and C_I at high frequencies, and the linear 45º slope is produced by the prevalence of Warburg impedance at low frequencies.

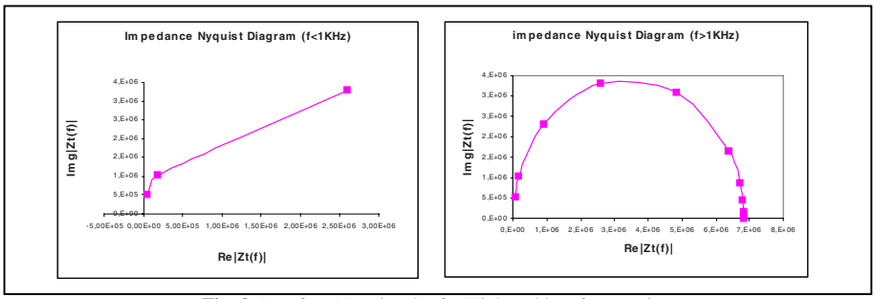


Fig. 3. Interface Nyquist plot for High and low frequencies.

Regarding the magnitude and phase response in frequency domain, this can be calculated from the model equation:

$$Z_T = R_S + \left[\frac{1}{C_I \omega j} // (R_{CT} + Z_W)\right] = \quad = R_S + \frac{1}{\omega_P C_I} \cdot \frac{1 + \dfrac{\omega}{\omega_P + 1/C_I R_S} j}{1 + \omega/\omega_P},$$

$$\omega_P = 1/\left(R_{CT} + Z_W\right) C_{dl}$$

For frequency considerations, we can use the superposition principle to separate the frequency spectrum into high and low frequencies. At high frequencies Z_w 0, then the Bode plot of Z_T shows a transfer function with one zero - one pole. The Z_o (low frequencies) will be $R_s + (1/\ \omega_P C_I)$. At low frequencies, Z_T $R_S + R_{CT} + Z_W$, and as $Z_W = K_W \cdot (j\omega)^{-1/2}$, The Bode plot is drawn as a –10dB/dec slope. The total Bode plot is shown in the next figure (Figure 4).

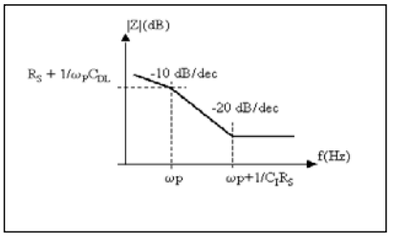


Fig. 4. Interface model Bode plot

3. Experimental Methods.

Electrochemical Impedance Spectroscopy (EIS) is a valuable technique widely used in different research areas. The method involves the application of a small sinusoidal perturbation in the injected signal at different frequencies. The response to this perturbation is measured and the magnitude of impedance and phase shift is determined. The equivalent circuits method was used to analyse the impedance spectrum. In order to characterize these parameters from a one hundred electrode array in saline solution, we built-up the following set-up (Figure 5).

The array was dipped into a ringer bath, with an Ag/AgCl reference electrode closing the circuit. A shunt resistance (R_{shunt}) of 150 Ω was added to give information about the current trespassing the interface. Sinusoidal current waveforms with varying frequency and amplitude were injected to study the impedance components, measuring phase shifts and amplitude changes between voltage and current.

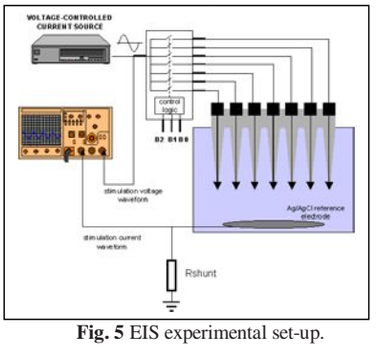


Fig. 5 EIS experimental set-up.

4. Results.

It is clearly seen from the results obtained that an increase in frequency arises an impedance (phase and modulus) decrease (Figure 6). This leads to a reduction in the amount of power delivered for a certain current level as higher frequencies are

reached. Although an impedance reduction could benefit from a lower charge to provide by a neuroprosthetic device, high frequencies could negatively affect nervous tissue. Figure 7 shows the Nyquists plots experimentally obtained at low and high frequencies combined for two different electrodes. It can be seen the circular structure for high frequencies and the linear component for the low ones, produced by the ionic diffusion in Warburg impedance models. Both graphics match the theoretical model plots shown in Figure 3.

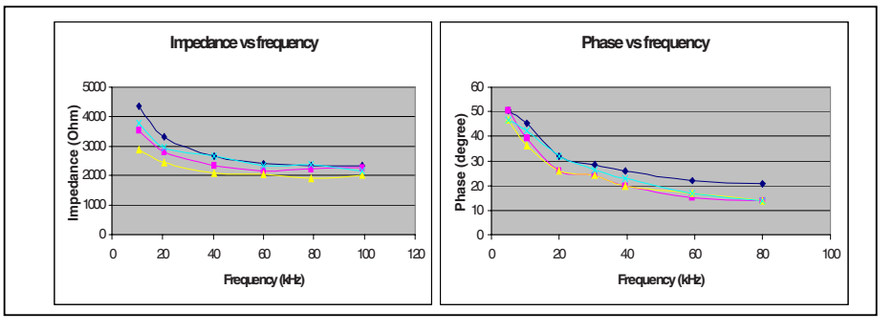


Fig. 6. Impedance modulus and phase vs. frequency.

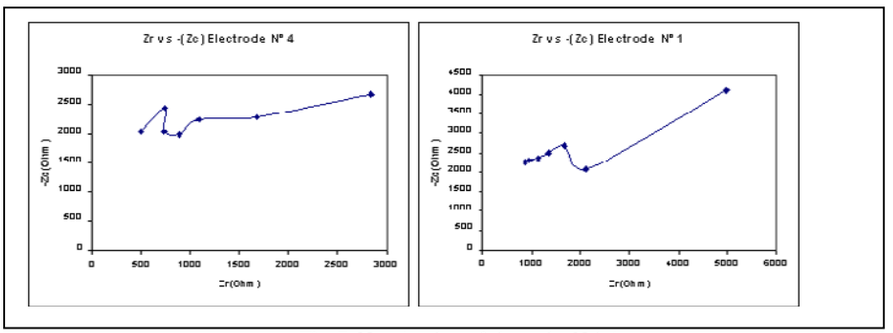


Fig. 7. Nyquist plots of electrodes N 1,4.

Actively balanced stimulating current waveforms are generally considered for a safe stimulation. The most commonly waveform used for stimulation is a Cathodic-First (CF) bipolar current pulse. This signal allows the re-polarization of the neural media in the second phase, therefore, it is easy to monitor the evoked spikes, minimizing stimulus artifacts. Moreover, it is more adequate to stimulate by means of constant current stimulus because it is more reliable to control the charge injection through any electrode-tissue interface. Because charge injection becomes independent of tissue impedances, an increase in safety is achieved. In the following Figure a CF current pulse is shown with its associated delivered voltage waveform. A predominant capacitive behavior is observed since voltage linearly decreases with the negative phase of the current waveform (A-B segment) and a consequent positive voltage slope appears while the positive current phase (C-D segment). The B-C segment evidences

the Warbourg resistance where charge transfers take place, forcing the current source to increment the delivered voltage in order to compensate the electronic ionic loss.

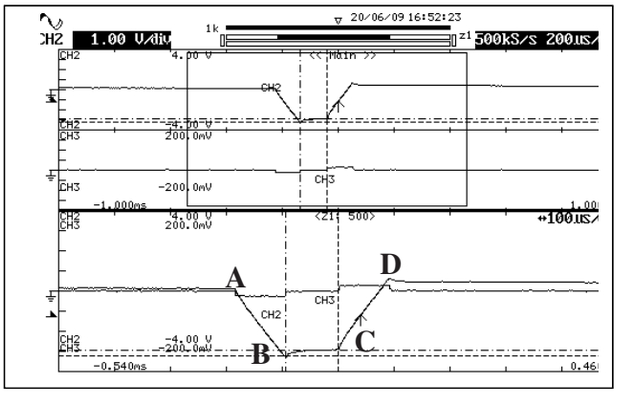


Fig. 8. 25 µA Cathodic-First bipolar current pulse (CH3). Voltage delivered waveform (CH2)

Unlike the results presented at Figure 8, for larger amounts of charge (100 µA), diffusion components become significant, therefore, a more resistive behaviour is expected. In Fig. 9 it can be seen how the voltage waveform follows the current waveform, which is telling the observer that there would be a mainly linear correlation between voltage and current obeying Ohms's Law. As an ideal stimulation scenario, the unique process to take place should be the Faradic one, but in this case, large surfaces would be needed. In fact, no pure Faradic or No-Faradic works appear, but a combination of them. Nevertheless, it is important to avoid irreversible zones. In the last case, a net charge transfer will be produced between electrode surface and electrolite, by means of no-reversible REDOX process, which liberates H_2 and Cl_2, modifying the pH condition in the interface. This may seriously compromise the electrode compatibility.

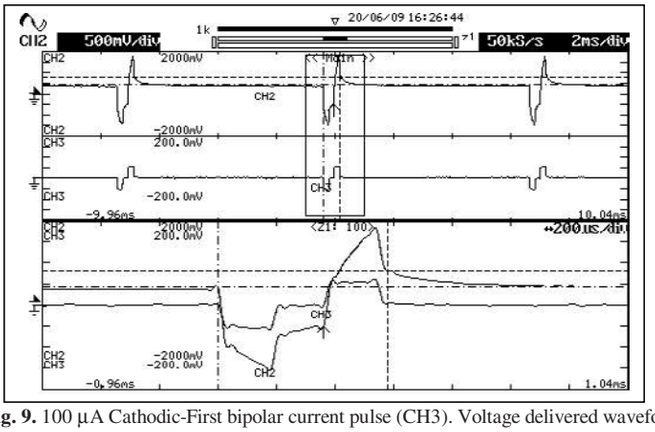


Fig. 9. 100 µA Cathodic-First bipolar current pulse (CH3). Voltage delivered waveform (CH2)

5. Conclusions.

In this paper we propose a model for characterizing a 3D intracortical electrode using Warbourg's model. It is shown the capacitive behaviour of the electrode denoted by a decrease in the impedance as frequency increases, according to the electrical model. The proposed model fits the expected behaviour, as can be denoted in the experimental Nyquist and Bode plots. Moreover, it is seen that as charge injection increases (100 μA) a resistive component appears, implying an irreversible Faradaic process which should be avoided in chronic stimulation since recombination of ionic species would appear in the surroundings of the electrode-tissue interface. Future work will be addressed to collect all the information required to characterize the impedance parameters in culture medium where large polar biomolecules effects will be studied.

Acknowledgements
This research is being funded by E.C. QLF-CT-2001-00279 and MAT2000-1049 to E.F. JM.F., FJ.G. and by F. Seneca OI-26/00852/FS/01 to JM.F.

References

1 Agnew, W.F. and McCreery D.B.: "Neural Prostheses. Fundamental Studies". Prentice Hall. New Jersey. 1990.

2 Heiduschka P.,Thanos S.:"Implantable Bioelectronic Interfaces for Lost Nerve Functions". Progress in Neurobiology 55, pp. 433-461, 1998

3 Schmidt E.M., Bak M.J. et al.: "Feasibility of a Visual Prosthesis for the Blind Based on Intracortical Stimulation of the Visual Cortex".Brain 119, pp. 507-522, 1996

4 Normann R., Maynard E., Guillory K., Warren D.: "Cortical Implants for the Blind". IEEE Spectrum May, pp.54-59, 1996

5 Blawas A.S.,Reichert W.M.: "Protein Patterning". Biomaterials 19, pp.595-609, 1998

6 Campbell P., Jones K., Huber R., Horch K. Normann R.: "A Silicon-Based, Three-Dimensional Neural Interface:Manufacturing Processes for an Intracortical Electrode Array". IEEE Transactions on Biomedical Engineering Vol.38 N 8 , pp.758-767, 1991

7 Mundt C., Nagle T., : "Applications of SPICE for Modeling Miniaturized Biomedical Sensor Systems". IEEE Transactions on Biomedical Engineering Vol.47 N 2 , pp.149-154, 2000

8 Ranck J.:"Which Elements are excited in Electrical Stimulation of Mammalian Central Nervous System :A Review". Brain Research 98 , pp.417-440, 1975

9 Newmann J.:"Resistance for flow of current to a disk", J. of Electrochemical society,113(5),:501-502,1966.

X d o d t o t o
D m o d t o t t

o o a a a M r o a o o ía o r a

D of Co u r rc c ur nd c no o y,
n n r nfor c , n r y of r n d , r n d , n
{acanas, eva, afdiaz, jortega}@atc.ugr.es

a or r n n P f d for rd n or o
d n on y r r y conn c d nd o r f ur , uc conf ur b
c on func on nd b c d b c ro on d ff r n oo n
o n u rn n od nd d u y r P rc
ron P b c u nd conn c y of P H r d
cr b rc c ur , r ou c on func on c n u , rn
n or , nd o b u of d cr on n nd d for rd r
n on o o conf ur b r c oo d o d o r n nd
u ny P n or o y or r y conn c d n y,
r n o r u on c r co n on n ord r o co r o nd r
conn c y, con nuou nd d cr o r on

u i

r r ra or or ar a rob o a a
o r b a a o a by a ara o T o ar
o b o ob a a o o b r o o ar ra or
M a b ro o a y or ora r
or a o ro y So o a a a o M
r o b r o ara r ra b r
ra a o ab y or ara a o
W a or a a a o ar ar r or a
ra o a a o *n L r P r r n* M a
ra o o or ar or a a a o o a
ay r a b o r a a r a a a o or a ro
a a o or a ora o or r a rob Ta abo
o o a o y ar a o y a So o r o a
o b y o bor oo r r o o a a b a a
or o o ar y W ar r o b o or
L r P r r n SM or *m D r r*
T W r ary r o o M or r o o o o o
a or a a a o ar ro ra o ra a r
y a oy o ra r o r
r a o o a a a o

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 89-96, 2003.

2 XMLP Architecture

The XMLP is a feed-forward neural network with an input layer (without neurons), a number of hidden layers selectable from 0 to 2, and an output layer. Besides the usual MLP connectivity, any layer can be two dimensional and partially connected with adjacent layers. As depicted in Fig. 1, connections come out from each layer in overlapped rectangular groups. The size of a layer and its partial connectivity are defined by six parameters in the following form: $x(g_x, s_x) \times y(g_y, s_y)$, where x and y are the sizes of the axes, and g and s specify the size of a group of neurons and a step between two consecutive groups, both in abscissas (g_x, s_x) and ordinates (g_y, s_y). A neuron i in the X axis at layer l (the upper one in Fig. 1) is fed from all the neurons belonging to i-th group in the X axis at layer l–1 (the lower one). The same applies simultaneously for Y axis. When g and s are not specified for a particular dimension, total connectivity is assumed for that dimension ($g_x = x$ and $s_x = 0$, or $g_y = y$ and $s_y = 0$). Thus, MLP is a particular case of XMLP where $g_x = x$, $s_x = 0$, $g_y = y$ and $s_y = 0$ for all layers.

3 XMLP Tool

An interactive tool (Fig. 2) has been developed both for SUN workstations and for PCs (running Linux), early using AXON neurosoftware from HNC and later C language. This tool allows the modification of numerous parameters in order to adapt to different problems. Users can configure the XMLP topology, select different operation modes (learning, test, production, etc.), activation functions and learning parameters, and obtain graphic results about learning, recognition rates, weights, etc.

4 Activation Functions

It is possible to choose between sigmoid, hyperbolic tangent, arc tangent and ramp activation functions. In order to achieve generality, the four functions have been

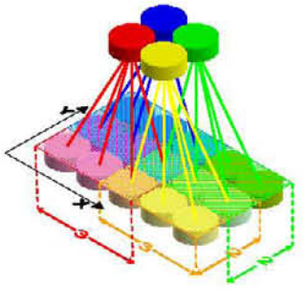
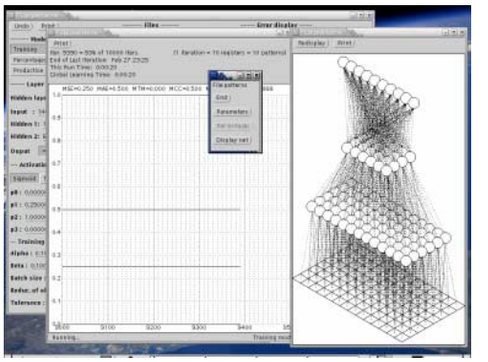


Fig. 1. Structure of an example XMLP layer, defined by the expression $5(3, 2) \times 3(2, 1)$, showing its connections to the next forward layer (on top)

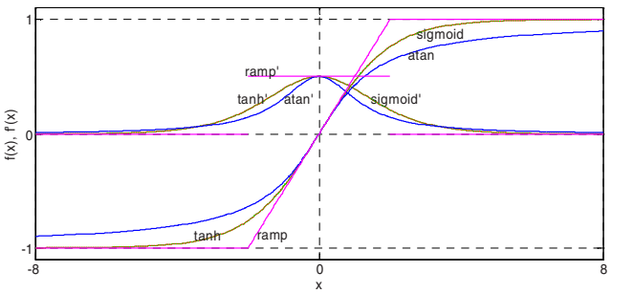
a o r ara r ′ o a a a o
o a a T r r o or o
a r r a ba ro a a o a or ar
Tab a or r ar ar a o ara r o



F P oo runn n on PC nu

a e c on func on nd r d r

gm	$= \frac{a - i}{-\frac{}{-i}} + i$ ′ a a ar o o +
	By $= \frac{}{-\frac{}{-i}}$ a ob a a r o o + a a o o ′ = ′ (−)
t n	$= \frac{-i}{-i}$ a $\frac{'}{-i} + \frac{+i}{}$ o o o
	$f'x = f'\left(1 - \frac{2}{ax^{-}}^{2}\frac{2'}{}x\right)$
t n	$= \frac{-i}{\pi}$ ar a $\left(\frac{\pi'}{-i}\right) + \frac{+i}{}$
	$' = \frac{'}{+\left(\frac{\pi'}{-i}\right)}$
mp	$\leq -\frac{-i}{,i}$ $= ' + \frac{+i}{}$ $-\frac{-i}{,i} < < \frac{-i}{,i}$ a $\geq \frac{-i}{,i}$
	$' = '$ $-\frac{-i}{,i} < < \frac{-i}{,i}$ o r



F c on func on nd r d r for f′ = , f_{ma} = , f_{min} =

a i

T M ba ro a a o r a ro ba a r a r a y o oo o T ar a by

ra o =

$$^{t} = \begin{cases} ^{t-} & - \text{ o } < - \\ ^{t-} + \Delta\ ^{t} & - \text{ o } = - \end{cases}$$

$$\Delta\ ^{t} = \begin{cases} \Delta\ ^{t-} & - \text{ o } < - \\ \text{S oo a } -\beta\ \alpha\Delta v^{t} + \beta\Delta\ ^{t-} & \\ \text{S oo b } \alpha\left(-\beta\ \Delta v^{t} + \beta\Delta\ ^{t-} \right) & - \text{ o } = - \\ \text{Mo } \alpha\Delta v^{t} + \beta\Delta\ ^{t-} & \end{cases}$$

$$\Delta = \Delta v^{t} = \begin{cases} \delta z & - \text{ o } = \\ \delta z + \Delta v^{t-} & - \text{ o } > \end{cases}$$

r o o o ro a ro ay r o a ro ay r *z* o o a ro δ ba ro a a rror r α a β ar r y a a oo o r r r a or ra o a o ba b r o o ra o o a

To r a ar r or a M oo a o oo b r o b y o a

n m or ar or a o a r o

n r r′ a or r o a a a a a ra *r′* aro

T ro ar b a o a r o r r a o a a o o r a y a o q r a r orr o o

n I ar a o a a ra a a a o o

T o r o M o o b r o ra o

i i a i

T M oo or ora o b y o or r a T o a a ar ar a o o r ar a o a y a o o a r a r r ra o ar ab orr o o M a a o o o bo a b r o b a a a o o a o ra by b r o b *n* o r ar ab a a a a o or r o o o ar ab T r o or o r o b o o a r a ar

$$= \mathrm{ro} \left(\frac{^{n-1}\_}{\mathrm{a}\left(\left| \; || \; _{M} \right|\right)} \right) = \frac{\mathrm{a}\left(\left| \; || \; _{M} \right|\right)}{^{n-1}\_}$$

o b o o ra o a a o o by o a o b a ro a o o a a ro a r a o r o o r a ab o o o a o M a a a o o o b or r a o a ab o o o a o b or r a o a b r o o a ro 6 T a ab o a o r b 6 b r o b ary o r r a a a y o ary o a b o or o oo ab a or r a a o o b o y o a a o a b o r r o ob a r a o ab a ra y W b or a a o o a b or o oo ab o a 6 r a o or a a b a ra o a o o o r aro or a T r o o a r ra o a b a o o a o a a o o o ar o oo ab r a o b r a ar ar a o a o a o a a a R M or a R M a o y or a ar ab or o

7 M a u a G

T M oo o o r a r Tab T r r ar ab o a y y o r o T a r a o y M or a a o a a a r o b orr o o r a a b

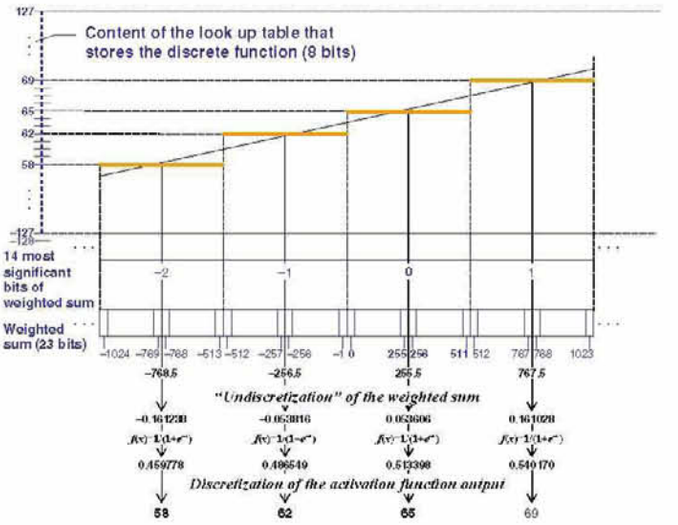


Fig. 4. Example of computation of the discrete sigmoid function

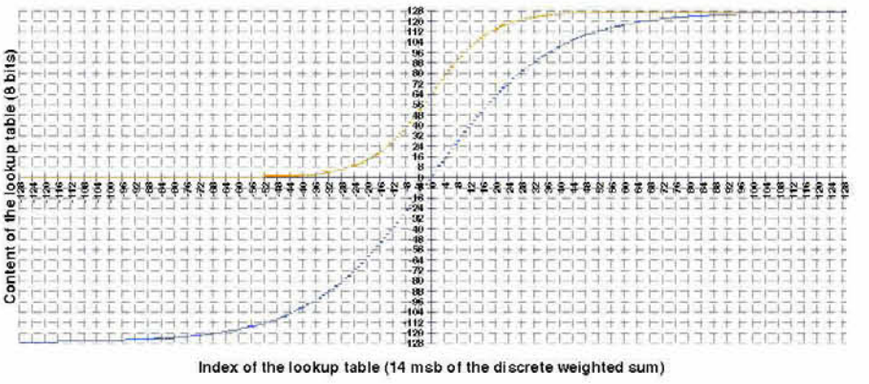


Fig. 5. Discretization of $1/(1+e^{-x})$ (upper plot) and $2/(1+e^{-x/2})-1$ (lower plot). The upper discrete function takes values different from –127 and 127 only in 0.94% of the entries

8 Results in Speech Recognition

Finally, we present results that compare totally and partially connected topologies (Table 3). The examples consist in the classification of the phonemes present in the spoken Spanish digits, and the recognition of the digits itself as isolated words.

Table 4 shows other results on discretization in the case of phoneme recognition.

a e ur of rror nd oodn for M r on nu b r of ou u , d r d ou u , y c u ou u , nd d r d c n c f c on

n *r* *rr r*	M = —— $\sum_{t=}^{N-}$ $\left\| \ t - t \right\|$ MS = —— $\sum_{t=}^{N-}$ $\left(\ t - t \right)$
n *r n*	MTM = — v_t v_t = $\left\| \ t - t \right\| \leq$ o ra $\forall$ ≤ ≤ o r
n *rr r*	M = —— $\sum_{t=}^{}$ $\left\| \ {}^{t}_{c} - {}^{t}_{c} \right\|$ + ——— $\sum_{=\neq c}^{N-}$ $\left\| \ t - t \right\|$
n *C rr* *C* *n*	M = — $\sum_{t=}^{}$ t t = $^{t}_{c} > {}^{t}$ ∀ ≤ ≤ ∧ ≠ o r
n R n *C rr* *C*	= —— $\sum_{t=}^{}$ t, $^{t} \geq \geq {}^{t} \geq \geq {}^{t}_{-1} \wedge i = c$

a e Co r on b n fu y conn c d P nd r y conn c d P ‘ ’, ‘ ’, ‘o’ nd for nu b r of n u , dd n n uron nd ou u n uron , r c y

rob	r ro	M	M
o R o o 6 o	r r y 6	× o 6 66 orr rror	× × o 6 6 orr rror
	6 T	× 6 o 6 6 6 6 orr rror	× 6 × o 6 6 6 orr 6 rror
o a Wor R o o S a	r Δ r r y Δ r y	× o 6 orr rror	× ×6 o 6 orr rror
	6 T	× 6 o 6 6 orr rror	× 6 ×6 o orr rror

u i

W a r a or ar ra or o a orr o a o oo o o a ay r ar a y o o rab or a ra o a a o R r o o o o ar a y o

Table 4. Comparison between continuous and discrete MLP

Problem	Continuous MLP	Discrete MLP
Phoneme Classification, preprocessing 16 FFT	inputs: 11×16, hidden: 32, outputs: 16, 6144 weights 69.33% correct 30.67% error	69.00% correct, 31.00% error inputs & function output: 8 bits weights: 8 bits, function input: 14 bits

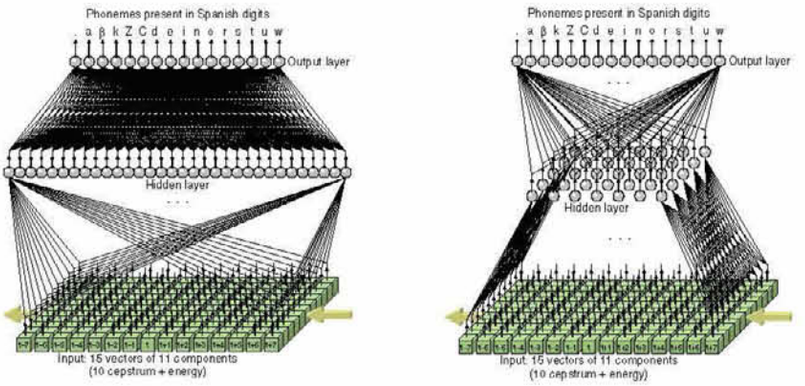


Fig. 6. MLP (left) and partially connected XMLP (right) for phoneme recognition

nected modular architectures can perform as well as fully connected ones, with a dramatic reduction in the number of weights and also in the learning time. Discrete feed-forward operation also gives comparable results to continuous one. These facts are very valuable for high-speed, low-cost hardware implementations, for example in FPGAs.

References

1. Caelli, T., Guan, L., Wen, W., Modularity in Neural Computing, Proceedings of the IEEE, Vol. 87, 9, 1497 -1518 (1999)
2. Krause, A., and Hackbarth, H., Scaly Artificial Neural Networks for Speaker-Independent Recognition of Isolated Words, Proc. of the IEEE Int. Conf. On Acoustics, Speech and Signal Processing, ICASSP '89, 21-24 (1989)
3. Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., Lang, K., Phoneme Recognition Using Time Delay Neural Networks, IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 37, No.3, 328-229 (1989)
4. Cañas, A.; Ortega, J.; Fernández, F. J.; Prieto, A.; Pelayo, F. J., An approach to isolated word recognition using multilayer perceptrons, in: Prieto, A. (Ed.): Artificial Neural Networks. LNCS 540, 340-347, Springer-Verlag (1991)
5. Fernández-Rodríguez, J. J., Cañas, A., Roca, E., Pelayo, F. J., Fernández-Mena, J., Prieto, A., Application of Artificial Neural Networks to Chest Image Classification, in: Mira, J., Cabestany, J., Prieto, A. (Eds.): New Trends in Neural Computation, LNCS 686, Springer Verlag (1993)

rr r r i
r f r r i y r i
r r

s l d z l t d s Á l í z d d[2] s [2]
z

D o c r n ca cno ogía o a ora ro c o
n r a o écn ca ar ag na n g o o a ar na
a ra a ar 0 0 ar ag na S IN
pe p t e
o ag http w et p p t e e pe ht
ra n o r In r In gr r Sc a ng n I D
o an
0 r ang n G r an

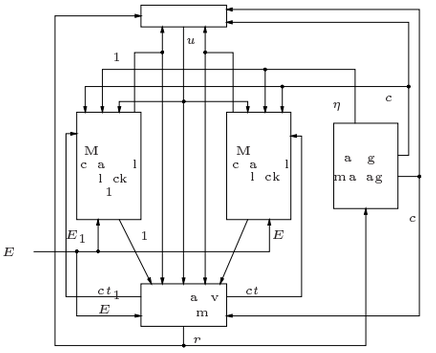
A st t In a r a x gna c rr n o c
n ng co rc a 0 c no og I r or r ro
c ng a on r n ron a r n R a n ra n
wor o a o a on ow an acc a n ar rror or c
n ra n gna w ng ro 0 o 0 c rc
o ra a a o ag o w 00 an w

n uc i n

S ss d t t d d d t s
l tw d l l s s d l l t t
s l ds l tw s s d t s d l s d
st w s 2 l tw l t t s tt s
zz zz 5 2 d s
t s l l tw s d s d d l d w d
l t s
l s tw s ll w d st t l
t s w l tw s l d s lw s d
ss s st l t t t l t s
d t t l t t s st s d s d
t s st s s s d S t st t st t s l
tw s l st ts ss ll l s s s lt t s t t
l t t s d s d s t d d s t st
t s s w l tw s l t d dw
S l s ls d s d l tw s s 7
d zz st
t s s t w t t dw l t t
s t d t s w s d t dw d s t

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 97-104, 2003.

d d s ss l s l s d l t t s s
t s s w w s l tw s
t t t s s t s t s z s t d l S st t s
s d s ts l wl d s t t
s s st t d s ll ws s t 2 d s s t s
ss l s t t d lt l s d l tw s s
l t t t d ts d l d s
s t s s s t s s ll w d t d s t t s t
d l t l ls t st ss l t ss s l
l l tw S l t s lts s w s t 5 w l
t l s d w d s ss d s t



F u oc agra o R n ra n wor

MA p c ssin y s

s l d d s t l tw t t s d d
d t d t t ss d l s t d s d s t d t
s st t t s l tw s s w s
ss l s t s l tw t t t s st t t
t w t ll l d s l l
l s d s s s d l t l l t t
s t t d t l z t s s t
s d l tw s Us ll t s l z t ss s d s 2
s l t s 2 t

2

,…,

,…,

s d s t t s 2 t s s t l t t
l z t d t st ss l t l
l tw

si n f sic p c ssin u s

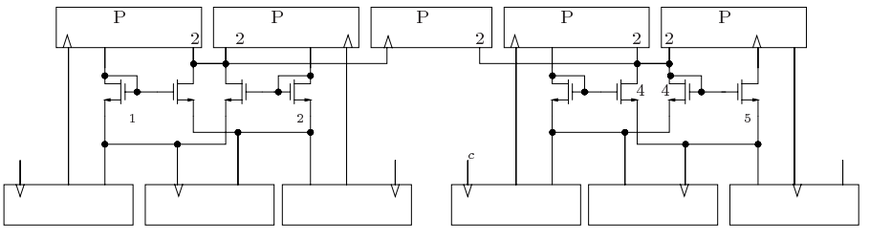
d t d l t w s t t s t s
2 d s l s ts st d l d s s s ts
ld l s t ll w

ss ll t t t t l 2
ll t d t t l
S st t d d t
d t t l z t

s ts d l d t d tt
d s l t t t d t s d d t
s t s s s ll t t lt d t ts s w ll
d s s d d d s l t ts ld l s
ll t t st s s d d t s l d
t s s t t d t t sl s d ts
s d t l t s s s t d t
t t l t s d d s t l t t s t t t
d s d d s dd t d s st t s l d st
l ff s t l w t t t d s

1 v er

d l t t d d t s t t s st d l d
t d d d s s t d s d d t ts
d s d l t d t t sl l t l
s t s st s Us t t tw t s t s
d d s s l t d 2 s ws t s l d s t d s
t lt l d d s d t s st s $_4$ s
t s t t w l l s d t s st s s
t s d d t
s t t ll t t s st s t st l $_4$ s —
$_2$ l l t t s l



F u 2 r r c a c gn

$\sqrt{\quad}$ $\sqrt{2}$ $\sqrt{\quad}$ $\sqrt{4}$ 5

s d d t s l d t s d ts t
t s t t l t s st s s t t t s st s t t
t s d d t s s — —— w l t s d d
t s s — ——

2 7

4 — — —

ll t t t t s t

$\sqrt{\quad}$

t s d t s st l t s t t s $\frac{5}{5}$ — d
— — $\frac{5}{5}$ l s s d 2 d W W
W W t d s t s l d d
s t s d s d d t s t s d ts
t t s t t t t t s t

— —
$\sqrt{\quad}$ $\sqrt{\quad}$ — —

o

2

o

o 2 —

d t t t st t s w 2 d t t s d
t t t t s l d t 5 d t t d

s t s t w t t t ts s t t st t
l

$$o \quad \text{——} \qquad 5$$

T T c rc

t s t t s d t s d s
w t t s l S *l* t s d t s
t s t t l s t ll t
t t s s st t d t t t t
d t — t s lt ts — s l t d t
t s t s st t t
t t d t s s d st — t t t t st
st t ts d s s d s l s s
t s d sw t s t st t

$$\text{—} \quad ,\ldots, \qquad 7$$
$$\text{—} \quad \text{—}$$
$$,\ldots, \quad \text{—}$$

d t t s l t tw s d d t t
t d s st t s ss s w ll w s t t
t d t t s d d d d l t
d l s d l d d t s s d t s l ss
t s st s t s z d d t t w t s l
t d l s s d t d t t t t
s w t t

$$\text{—} \qquad 2$$

s t t t t l t s t d ff tw t
d t t t s t ts t t t
s d ff s t d d l s s t s t t st
s t d s t l t s t d tl ff s t d d l
d d t l l w s t t t s d s
l t t s s w

p s si n

l tw l d l t l s s ll t s d l s d
s d t ss t s t t s d d d

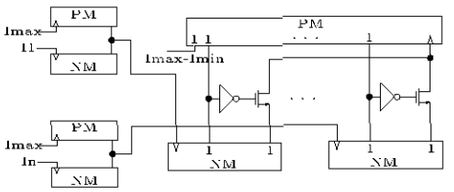


Figure 3. Schematic of the complementary LTA circuit

to send the proper currents to each one of the modules. The final design of the IC is shown in Figure 4. For testing purposes our design has been limited to five input signals $E_1 \ldots E_5$ and five ouput signals $N_1 \ldots N_5$

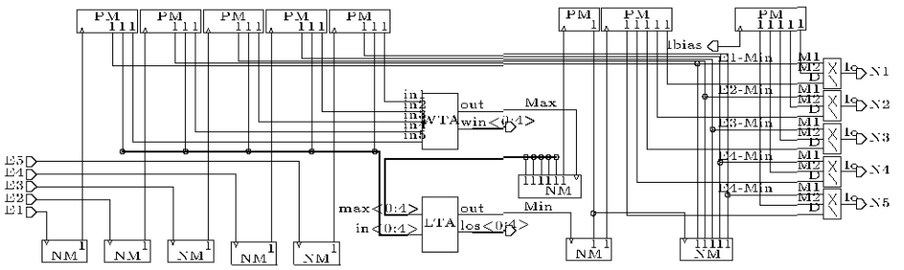


Figure 4. Schematic of the neural network layer designed

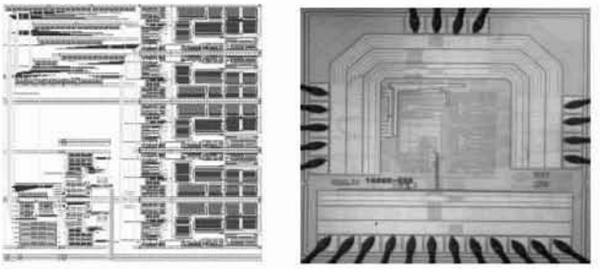
A prototype called CNNA_T1 has been developed for testing purposes using AMS 0.35μm commercial technology. The power suply and current swing are summarized in table 1. Layout size is 1.3 mm x 1.3 mm. Power dissipation is lower than 20 mW. Circuit bandwidth is 200 kHz.

	Analogue	Digital
Vdd	3.3V	3.3V
E_i	20–50μA	–
N_i	0–10μA	–

Table 1. Electrical specifications of CNNA_T1

5 Results

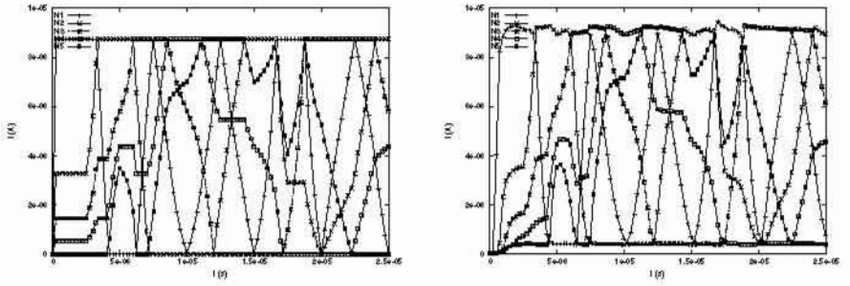
To test the performance of the design, several simulations have been performed. The simulation conditions are the following: triangular waveform input signal



(a) Design layout (b) Chip microphotograph

Figure 5. Layout of the F1 layer in the MART neural

with periods from 5 μs to 25 μs applied to $E_1 \ldots E_5$, respectively; 35μA bias current applied to I_{bias} and NMOS transistors in diode configuration, with aspect ratio of $\frac{W}{L} = \frac{10}{1}$ as output loads. The post–layout simulated output compared to the theoretical one has a deviation lower than 10%, which is acceptable for neural networks where the learning process can solve those deviations.



(a) Theoretical output (b) Post layout simulated output

Figure 6. Response comparison for the CNNA_T1 chip

6 Conclusions

This paper describes the design of CNNA_T1 chip, a neural network preprocessing layer which involves analogue operations such as current–mode divition, substraction, WTA–MAX and WTA–MIN. This layer can be found in ART–

s d l tw s s s lt l l tw t
t t s t s l d s t d l t l l
d s t s s w s w ll s s tw t s s
t d ts t t l
s l t d l w t t t d s ls t 2 z
w t t sw 2 t 5 d t t sw t
t d l ts s d t d s t w l l
tw t s l t s tt d w d
t d t t t sw s t s w ll

Ac n w n s

s s w s d t s t t d s d d
st t t t t S lt l d t t
U 2 2 t U s d d l t d t

f nc s

Ga ar n r an S n Gro rg R o a a a rn r cogn
on a organ ng n ra n wor *e*
Ga ar n r an S n Gro rg a ara arc c r or a
organ ng n ra a rn r cogn on ac n *e n g a c an*
age ce ng
G ar n r S Gro rg an J R no R S r r a
arn ng an c a ca on o non a onar a a a organ ng n ra
n wor *e a e*
Ga ar n r an S n Gro rg R a a arn ng an
ca gor a on o ana og a rn an a a r onanc *e a e*

G ar n r S Gro rg J ar on N an R no an D Ro n
R n ra n wor arc c r or ncr n a r arn
ng o ana og n ona a *an ac n n ne a ne*

Ga ar n r an S n Gro rg R S organ ng o a ca
gor r cogn on co or ana og n a rn *e c* 0
D c r
S rrano Go arr ona an nar arranco n R croc an
n R *an ac n n e a e*

o n r an G a w n rg x o SI n a on
o R *n e na na Sy n c an Sy e*
rnán D ga o *na n e a a ec a e c ac ón ne na*
y a cac ón a e e n e a n e gen e e n zac ón e ac en e
D n r a San ago o o a
0 R J g r n *na y an yn e S an nea c c* w r
S rrano Go arr ona an nar arranco g r c on c rr n o
w a ax c rc w c ca a *na S S a e c*
0

On the effects of dimensionality on data analysis with neural networks

M. Verleysen[1], D. François[2], G. Simon[1], V. Wertz[2*]

Université catholique de Louvain
[1] DICE, 3 place du Levant, B-1348 Louvain-la-Neuve, Belgium
[2] CESAME, 4 av. G. Lemaitre, B-1348 Louvain-la-Neuve, Belgium
{verleysen, simon}@dice.ucl.ac.be, {francois, wertz}@auto.ucl.ac.be

Abstract. Modern data analysis often faces high-dimensional data. Nevertheless, most neural network data analysis tools are not adapted to high-dimensional spaces, because of the use of conventional concepts (as the Euclidean distance) that scale poorly with dimension. This paper shows some limitations of such concepts and suggests some research directions as the use of alternative distance definitions and of non-linear dimension reduction.

1. Introduction

In the last few years, data analysis has become a specific discipline, sometimes far from its mathematical and statistical origin, where understanding of the problems and limitations coming from the data themselves is often more valuable than developing complex algorithms and methods. The specificity of modern data mining is that *huge* amounts of data are considered. There are new fields where data mining becomes crucial (medical research, financial analysis, etc.); furthermore, collecting huge amount of data often becomes easier and cheaper.

A main concern in that direction is the *dimensionality* of data. Think of each measurement of data as one observation, each observation being composed of a set of variables. It is very different to analyze 10000 observations of 3 variables each, than analyzing 100 observations of 50 variables each! One way to get some feeling of this difficulty is to imagine each observation as a point in a space whose dimension is the number of variables. 10000 observations in a 3-dimensional space most probably form a structured shape, one or several clouds, from which it is possible to extract some relevant information, like principal directions, variances of clouds, etc. On the contrary, at first sight 100 observations in a 50-dimensional space do not represent anything specific, because the number of observations is too low.

Nevertheless, many modern data *have* this unpleasant characteristic of being high-dimensional. And despite the above difficulties, there *are* ways to analyze the data,

* MV is a Senior research associate at the Belgian FNRS. GS is funded by the Belgian FRIA. The work of DF and VW is supported by the Interuniversity Attraction Pole (IAP), initiated by the Belgian Federal State, Ministry of Sciences, Technologies and Culture. The scientific responsibility rests with the authors.

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 105-112, 2003.

and extract information from observations. If the current data analysis methodologies are not adapted to high-dimensional, sparse data, then it is our duty to develop adapted methods, even if some well-admitted concepts must be questioned. In particular, artificial neural network methods, now widely and successfully used in data analysis, should be faced to high-dimensional data and modified if necessary.

This paper makes no pretence of presenting generic solutions to this problem; the current state-of-the-art is far from that. However, we will illustrate some surprising facts (Section 2) about high-dimensional data in general, and about the use of neural networks in this context (Section 3). In particular, we will show that the use of standard notions as the Euclidean distance, the nearest neighbor, and more generally similarity search, is not adapted to high-dimensional spaces. There is thus a need for alternative solutions; this paper only gives paths to future developments (Section 4), to a new research activity that could influence considerably the field of neural networks for data mining in the next few years.

2. Some weird facts about high-dimensional space

High dimensional spaces do in fact escape from our mental representations. What we take for granted in dimension one, two or three, because we can figure it out quite easily, might not actually hold in higher dimensions. Let's highlight some weird facts.

2.1. The empty space phenomenon

Scott and Thompson [1] first noticed some counter-intuitive facts related to high dimensional Euclidean spaces, and described what they called the "empty space phenomenon".

Fact 1. The volume of a hyper-sphere of unit radius goes to zero as dimension growths. The volume of a sphere of radius r in d dimensions is given by:

$$V(d) = \frac{\pi^{d/2}}{\Gamma(d/2+1)} r^d \,. \tag{1}$$

Figure 1a shows the volume for $r = 1$; we see that the volume rapidly decreases with d. So, in higher dimension, a unit sphere is nearly empty.

Fact 2. The ratio between of the volumes of a sphere and a cube of same radius tend towards zero with increasing dimension as illustrated in Figure 1b. In one dimension these volumes are equal, and in two dimensions the ratio is approximately 0.8, but in higher dimension we can say that the volume of a hyper-cube concentrate in its corners.

Fact 3. The ratio of volume of a sphere of radius 1 and 1-ε tend towards zero given the obvious fact that its value is equal to (1-ε) to the power d. With d as small as 20, and $e = 0.1$, only 10% of the original radius contains 90% of the volume of the outer sphere, and so the volume of it concentrates in an outer shell. The same holds for hyper-cubes and hyper-ellipsoids as well.

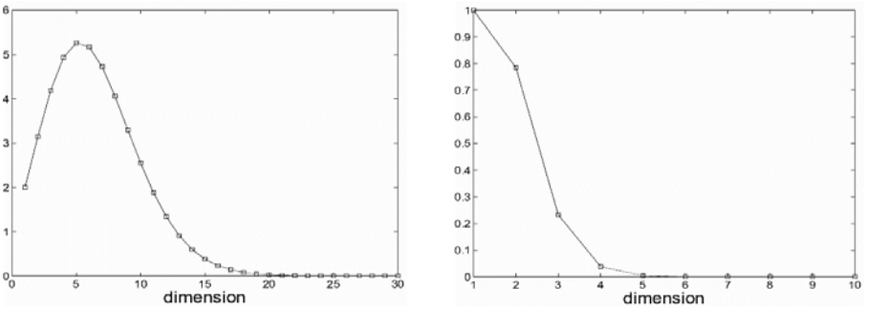


Fig. 1. Left (a): volume of the unit sphere; Right (b): ratio between the volumes of the unit sphere and the unit cube, with respect to the dimension of the space.

These observations imply that high-dimensional spaces are mostly empty. They indeed show that local neighborhoods of points are mostly empty, and that even in the case of uniform distributions, data is concentrated at the borders of the volume of interest.

2.2. The concentration of measure phenomenon

We will now have a deeper look at the behavior of the widely used Euclidean distance (i.e. the L_2-norm of the difference) when applied to high dimensional vectors.

Fact 1. The standard deviation of the norm of random vectors converges to a constant as dimension increases though the expectation of their norm growths as the square root of the dimension. More precisely, it has been proven in [2] that under i.i.d. assumption on x_i,

$$\mu_{\|x\|} = \mathrm{E}\left(\|x\|\right) = \sqrt{an - b} + \mathrm{O}(1/n) \tag{2}$$

$$\sigma^2{}_{\|x\|} = \mathrm{Var}\left(\|x\|\right) = b + \mathrm{O}\left(1/\sqrt{n}\right) \tag{3}$$

where a and b are constants depending only on the four first momentums of x_i.

Note that the same law applies to the Euclidean distance between any two points, since it happens to be a random vector too.

Fact 2. The difference between the distances of a randomly-chosen point to its furthest and nearest neighbor decreases as dimensionality increases. This can be illustrated by the asymptotic behavior of the relative contrast [3] :

$$\text{If} \quad \lim_{d\to\infty} \mathrm{Var}\left(\frac{\|x\|_k}{E\left(\|x\|_k\right)}\right) = 0 \quad \text{then} \quad \frac{Dmax^k - Dmin^k}{Dmin^k} \underset{p}{\to} 0 \tag{4}$$

where D_{min} and D_{max} are the distance to respectively the nearest and furthest neighbors of a particular point. Note that the hypothesis of the theorem is induced by equation (3). A more general proof of the theorem can be found in [4].

The conclusion we can draw from these observations is that, in high dimensional spaces, all points tend to be equally distant from each others, with respect to the Euclidean distance. As dimension increase, the observed distance between any two points tends towards a constant. This can be illustrated when computing the histograms of distances between random points of increasing dimensionality. It appears that (1) the mean of the histogram growths and (2) its variance shrinks.

2.3. The curse of dimensionality

Finally let us have a look at a not-so-weird-but-often-ignored fact, which Richard Bellman named "The Curse of Dimensionality" [5]. It refers to the huge amount of points that are necessary in high dimensions to cover an input space, for example a regular grid spanning a certain portion of the space. When filling an hypercube in 5 dimensions ($[0\ 1]^5$) with a 0.1-spaced grid, one needs no less than 100.000 points.

3. Consequences for neural network learning

The considerations developed in the previous section have important consequences on ANN (Artificial Neural Networks) learning. The following subsections give examples of such consequences in specific contexts.

3.1. Supervised learning

When modeling some process producing an output on basis of observed values for particular inputs, one has to fit a chosen model to a dataset. The more extensive the dataset, the more accurate is the model. Ideally, the dataset should span the whole input space of interest, in order to ensure that any predicted value (i.e. output of the model) is the result of an interpolation process and that no hazardous extrapolation occurs.

But one has to face the curse of dimensionality. Silverman [6] addressed the problem of finding the necessary number of training points (samples) to approximate a Gaussian distribution with fixed Gaussian kernels. His results show that the required number of samples grows exponentially with the dimension. Fukunaga [7] obtained similar results for the k-NN classifier showing that whereas 44 observations are sufficient in 4 dimensions, not less than $3.8e^{57}$ are necessary when dimension is 128.

3.2. Local models

Local artificial neural networks are often argued to be more sensitive to dimensionality than global ones. By local models, we mean approximators (or classifiers, or density estimators) made of a combination of local functions (for example Gaussian kernels). Indeed Gaussian functions also have an unexpected

behavior when extended to high-dimensional spaces. Examples of such approximators are RBFN (Radial-Basis Function Networks) and kernel methods.

When a normal distribution with standard deviation σ is assumed, the probability density function to find a point at distance r from the center of the distribution is given by [8] :

$$f(r) = \frac{r^{d-1}}{2^{d/2-1}} \frac{e^{-r^2/2\sigma^2}}{\Gamma(d/2)}, \tag{5}$$

which is maximum for $r/\sigma = (n\text{-}1)^{0.5}$. In one dimension, it is maximum at the center of the distribution, as expected, but when dimension growths, it diverges from the center (see Figure 2), which become nearly empty, whereas the Gaussian distribution is maximal ! This shows that Gaussian kernels are not local any more in higher dimensions, and that models that have been seen as sums of local kernels do not behave as such in high dimensions.

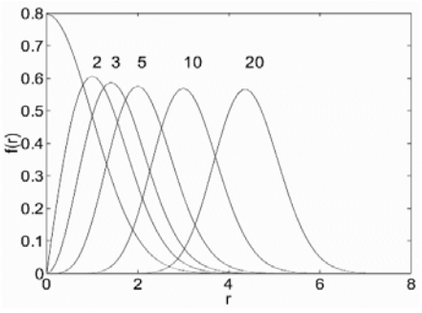


Fig. 2. Probability density of a point from a normal distribution to be at distance *r* of the center, for several space dimensions.

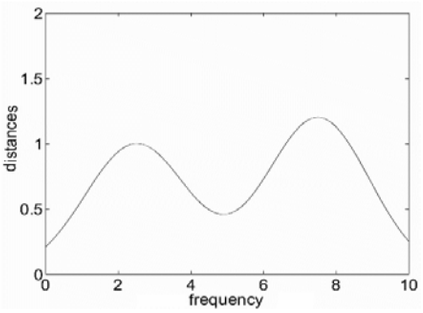


Fig. 3. Example of distance histogram in a several-clusters distributions.

This limitation to the use of local models, in particular with Gaussian kernels, seems severe. However, it should be emphasized on the fact that global models, as for example MLP (Multi-Layer Perceptrons), probably equally suffer. Indeed, in many cases, sums of sigmoids as in MLP result in functions taking significant values in a limited region of the spaces. While mathematically different, models as MLP and RBF thus often behave similarly in practice. This enforces the conviction that local and global models equally suffer from the curse of dimensionality (and related effects), while this is probably harder to prove for global models.

3.3. Similarity search and Euclidean distances

Most neural network models, as well as clustering techniques, rely on the computation of distances between vectors. For RBFN, it is the distance between a data and each kernel center. For MLP, it is the scalar product between a data and

each weight of the input layer. Both these distance measures may be related to the similarity search in clustering techniques, also used in vector quantization, LVQ, Kohonen maps, etc. Similarity search consists in finding in a dataset the closest element to a given point. In the context of clustering for example, efficient clustering is achieved when data in a cluster are similar (i.e. close with respect to the distance function) and data in different clusters are far away from each other. So, when data contain clusters, distance histograms should ideally show two peaks (as in Figure 3) : one for intra-cluster distances, and the other for extra-cluster distances. But if the distance histogram only contains one peak, or if the two peaks are close, distance-based clustering will be difficult. Unfortunately, the fact is that in high dimensions, any distance histogram tends towards a more and more concentrated peak, making the clustering task uneasy [9]. This is a direct consequence of the concentration of measure phenomenon.

4. Towards solutions

Effects of the curse of dimensionality and related limitations on neural network learning seem unavoidable in high-dimensional spaces. There are however at least two paths to explore to remedy to this situation.

4.1. Alternative distance measures

The use of the Euclidean distance between data is conventional and is rarely discussed. However, it is not obvious that another definition of distance could not be more appropriate in some circumstances, and in particular in high-dimensional spaces. In practice, any distance measure between vectors x and y (with components x_i and y_i) of the following form could be considered:

$$\|x - y\|_r = \sqrt[r]{\sum_{i=1}^{d} |x_i - y_i|^r} \quad . \tag{6}$$

In practice, (4) is applicable for any positive value of r; the asymptotical behavior of any distance definition as (6) is the same (i.e. all distances are subject to the concentration phenomenon). But the convergence rate of (4) differs for different values of r.

The intuition tells that using high values of r can mitigate the effects of loss of locality for Gaussian-like kernels. Nevertheless it has been shown [3] that lower values of r can keep the relative contrast (4) high (for a particular dimension). Unfortunately there is no known reason (other than numerical computation-related arguments) to find a lower bound for optimal r. And there is no sense in taking r = 0...Therefore it remains to find the proper to set a lower bound for r, so that an optimal and most probably dimension-dependant compromise can be found.

4.2. Non-linear projection as preprocessing

Another way to limit the effects of high dimensionality is to reduce the dimension of the working space. Data in real problems often lie on or near submanifolds of the input space, because of the redundancy between variables. While redundancy is often a consequence of the lack of information about which type of input variable should be used, it is also helpful in the case where a large amount of noise is unavoidable on the data, coming for example from measures on physical phenomena. To be convinced of this positive remark, let us just imagine that the *same* physical quantity is measured by 100 sensors, each of them adding independent Gaussian noise to the measurement; averaging the 100 measures will strongly decrease the influence of noise on the measure! The same concept applies if n sensors measure m quantities ($n > m$).

Projection of the data on submanifolds may thus help. A way to project data is to use the standard PCA (Principal Component Analysis). However PCA is linear; in most cases, submanifolds are not linear (think for example to a horseshoe distribution, as in [10]) and PCA is not efficient.

Alternative nonlinear methods exist to project data in a nonlinear way. Examples are Kohonen self-organizing maps (usually to project data onto one- or two-dimensional spaces), and methods based on distance preservation. The latter include Multi-dimensional scaling [11-12], Sammon's mapping [13], Curvilinear Component Analysis [14] and extensions [15]. All these methods are based on the same principle: if we have n data points in a d-dimensional space, they try to place n points in the m-dimensional projection space, keeping the mutual distances between any pair of points unchanged between the input space and the corresponding pair in the projection space. Of course, having this condition strictly fulfilled is impossible in the generic case (there are $n(n-1)$ conditions to satisfy with nm degrees of freedom); the methods then weight the conditions so that those on shorter distances must be satisfied more strictly than those on large distances. Weighting aims at conserving a local topology (locally, sets of input points will resemble sets of output points).

An example of successful application of the above approach in the context of financial prediction can be found in [16].

5. Conclusion

Theoretical considerations show that using classical concepts in data analysis with neural networks to process high-dimensional data may be not appropriate. The reason is that some of the underlying hypotheses, though obvious in lower dimension, are not verified any more in higher dimensions. Indeed, in practice, one observes severe performance loss with data processing algorithms when data are high dimensional. There is thus a need to adapt our models to high dimensionality. A way one can think of is to consider new similarity measures between data, other than the ancestral Euclidean distance. Another way is to reduce the dimension through projection on

(non-linear) submanifolds. In both cases, deep investigation is required in order to successfully adapt data processing tools to high dimensional data.

References

1. Scott, D.W., Thompson, J. R.: Probability density estimation in higher dimensions. In: Douglas, S.R. (ed): Computer Science and Statistics. Proceedings of the Fifteenth Symposium on the Interface, North Holland-Elsevier, Amsterdam, New York, Oxford (1983) 173–179
2. Demartines, P. : Analyse de données par réseaux de neurones auto-organisés. Ph.D. dissertation (in French), Institut National Polytechnique de Grenoble - France (1994)
3. Aggarwal, C. C., Hinneburg, A., Keim, D. A.: On the surprising behavior of distance metrics in high dimensional spaces. In: Van den Bussche, J., Vianu, V. (eds): Proceedings of Database Theory - ICDT 2001, 8th International Conference Lecture Notes in Computer Science, vol 1973. Springer, London, UK (2001) 420–434
4. Beyer K. S., Goldstein, J., Ramakrishnan, R., Shaft, U.: When is "nearest neighbor" meaningful? In: Beeri, C., Buneman, P. (eds) : Poceedings of Database Theory - ICDT '99, 7th International Conference. Lecture Notes in Computer Sciences, vol 1540, Springer, Jerusalem, Israel (1999) 217–235
5. Bellmann, R.: Adaptive Control Processes: A Guided Tour. Princeton Univ. Press (1961)
6. Silverman, B.W.: Density estimation for statistics and data analysis. Chapman & Hall (1986)
7. Fukunaga K.: Introduction to Statistical Pattern Recognition. Academic Press, Boston, MA, (1990)
8. Hérault, J., Guérin-Dugué, A., Villemain, P.: Searching for the embedded manifolds in high-dimensional data, problems and unsolved questions. Proceedings of ESANN'2002 - European Symposium on Artificial Neural Networks, d-side public, Bruges - Belgium (2002) 173–184
9. Steinbach, M., Ertoz, L., Kumar, V.: Challenges of clustering high dimensional data. New Vistas in Statistical Physics – Applications in Econo-physics, Bioinformatics, and Pattern Recognition, Springer-Verlag (2003)
10. Verleysen, M.: Learning high-dimensional data. Acc. for public. in Ablameyko, S., Goras, L., Gori, M., Piuri, V. (eds): Limitations and future trends in neural computation, IOS Press.
11. Shepard, R. N.: The analysis of proximities: Multidimensional scaling with an unknown distance function, parts I and II, *Psychometrik*a, 27 (1962) 125-140 and 219-246
12. Shepard, R.N, Carroll, J.D: Parametric representation of nonlinear data structures. In P. R. Krishnaiah (ed.): *International Symposium on Multivariate Analysi*s, Academic Press, (1965) 561-592
13. Sammon, :A nonlinear mapping algorithm for data structure analysis, *IEEE Trans. on Computer*s, C-18 (1969) 401-409
14. Demartines, P., Hérault, J.: Curvilinear Component Analysis: a self-organizing neural network for nonlinear mapping of data sets, *IEEE T. Neural Network*s,. 8-1 (1997) 148-154
15. Lee, J. A., Lendasse, A., Verleysen, M: Curvilinear Distance Analysis versus Isomap. In: Proceedings of ESANN'2002, 10th European Symposium on Artificial Neural Networks, d-side public, Bruges – Belgium, (2002) 185-192
16. Lendasse, A., Lee, J. A., de Bodt, E., Wertz, V., Verleysen, M.: Dimension reduction of technical indicators for the prediction of financial time series - Application to the Bel 20 market index. *European Journal of Economic and Social Systems*, 15-2 (2001), pp. 31-48

Hardware Optimization of a Novel Spiking Neuron Model for the POEtic tissue.

Oriol Torres[1], Jan Eriksson[2], Juan Manuel Moreno[1], Alessandro Villa[2,3]

[1]Technical University of Catalunya, Barcelona, Spain
[2]Lab. of Neuroheuristics, University of Lausanne, Lausanne, Switzerland
[3]Inserm U318, Lab. Neurobiophysique, Université Joseph Fourier, Grenoble, France

Abstract. In this paper we describe the hardware implementation of a spiking neuron model, which uses a spike time dependent synaptic (STDS) plasticity rule that allows synaptic changes by discrete time steps. For this purpose it is used an integrate-and-fire neuron with recurrent local connections. The connectivity of this model has been set to 24-neighbour, so there is a high degree of parallelism. After obtaining good results with the hardware implementation of the model, we proceed to simplify this hardware description, trying to keep the same behaviour. Some experiments using dynamic grading patterns have been used in order to test the learning capabilities of the model.

1 Introduction

Artificial neurons have been modelled in the last years trying to characterise the most important features of real neurons and to achieve further aspects of computation like parallelism and learning. In this way, biologically-inspired neuron models, so-called neuromimes, have become an important path to follow up, due to their big potential in developing timing features of neural computation [1].

A commonly used model for neuromimetic studies is the Integrate-And-Fire model, which incorporates the integration of the input signals, until it overcomes a certain threshold and then the neuron fires [2][3]. In this model, we use a Spike Time Dependent Plasticity (STDP) rule, which allows to change the weight of the synapses depending on the time between spikes [4][5].

In the next chapter, we shall present in detail the Integrate-And-Fire model we use in our work considering both timing features and its novel learning rule. In chapter 3, its basic hardware implementation is developed, but due to its high complexity, in section 4 its optimisation is analysed. Finally, chapter 5 summarizes the conclusion and our future work.

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 113-120, 2003.

2 Neuron Model

This model consists basically in an Integrate-And-Fire scheme, in which synapses can change their weights depending on the time difference between spikes. This model has been developed within the framework of the POEtic project [6]. The outputs of the synapses are added until their result $V_i(t)$ overcomes a certain threshold θ. Then a spike is produced, and the membrane value is reset.

The simplified equation of the membrane value is:

$$Vi(t+1) = \begin{cases} 0 & \text{when } S_i(t)=1 \\ k_{mem} * V_i(t) + \sum J_{ij}(t) & \text{when } S_i(t)=0 \end{cases} \quad (1)$$

Where $k_{mem}=\exp(-\Delta t/\tau_{mem})$, $Vi(t)$ is the value of the membrane, J_{ij} is the output of each synapse and $S_i(t)$ is the variable which represents when there is a spike.

2.1 Synapse Model

The goal of the synapse is to convert the spikes received from another neurons in proper inputs for the soma. When there is a spike in the pre-synaptic neuron, the actual value of the output J_{ij} is added to the weight of the synapse multiplied by its activation variable. But if there is no pre-synaptic spike then the output J_{ij} is decremented by the factor k_{syn}.The output J of the synapse i-j is ruled by:

$$J_{ij}(t+1) = \begin{cases} J_{ij}(t) + (w_{RiRj} * A_{RiRj}(t)) & \text{when } S_j(t)=1 \\ k_{syn} * J_{ij}(t) & \text{when } S_j(t)=0 \end{cases} \quad (2)$$

where j is the projecting neuron and i is the actual neuron. R is the type of the neuron : excitatory or inhibitory, A is the activation variable which controls the strength of the synapse, and k_{syn} is the kinetic reduction factor of the synapse.If the actual neuron is inhibitory, this synaptic kinetic factor will reset the output of the synapse after a time step, but if the actual neuron is excitatory, it will depend on the projecting neuron. If the projecting neuron is excitatory the synaptic time constant will be higher than if it is inhibitory. The weight of each synapse also depends on the type of neuron it connects. If the synapse connects two inhibitory neurons, the weight will always be null, so an inhibitory cell can not influence another inhibitory cell. If a synapse is connecting two excitatory neurons, it is assigned a small weight value. This value is higher for synapses connecting an excitatory neuron to an inhibitory one, and it takes its maximum value when an inhibitory synapse is connected to an excitatory cell.

2.2 Learning Model

In order to strengthen or weaken the excitatory-excitatory synapses, the variable A will change depending on an internal variable called L_{ij} which is ruled by:

$$L_{ij}(t+1)=k_{act}*L_{ij}(t) + (YD_j(t)*S_i(t)) - (YD_i(t)*S_j(t)) \quad (3)$$

Where k_{act} is the kinetic activity factor, which is the same for all the synapses.

YD is the learning variable that measures, with its decayment, the time separation between a pre-synaptic spike and a post-synaptic spike. When there is a spike, YD will have its maximum value in the next time step, but when there is not, its value will be decremented by the kinetic factor k_{learn}, which is the same for all synapses

When a pre-synaptic spike occurs just before a post-synaptic spike, then the variable L_{ij} increases and the synapse strengthens. This means it reinforces the effect of a pre-synaptic spike in the soma. But when a pre-synaptic spike occurs just after a post-synaptic spike, the variable L_{ij} decreases, the synapse weakens and the effect of a pre-synaptic spike in the soma will descend.

3 Hardware Implementation

This section describes the digital hardware implementation of these models.

3.1 Neuron Block

First of all, when the input spikes , *s*, are received from another neurons, they are processed through a block that comprises two sub-blocks (synapse & learning), which are explained in the next sub-sections.

These inputs are added or subtracted, depending on the nature, called *r*, of the previous neuron (i.e. excitatory or inhibitory), to the decayed value of the membrane. When the registered membrane value, V_i, overcomes a certain threshold, V_{th}, a spike, S_i, is produced. This spike will be delayed in the final part with a chain of registers which model the refractory time, t_{refr}. When finally the spike goes out from the neuron, it produces a reset, *rst*, in the register which stores the value of the membrane. The block diagram of the neuron block is represented in fig. 1. The membrane path has a resolution of 12 bits, with a range [-2048, 2047], and the threshold, V_{th}, is kept fixed to +640.The membrane decay function has a time constant value of τ=20.The refractory time is set to 1. The decayment block is modelled following the implementation presented in [3]. This block depends on two output parameters and the MSB of the input, which establish the exponential decay required at the output

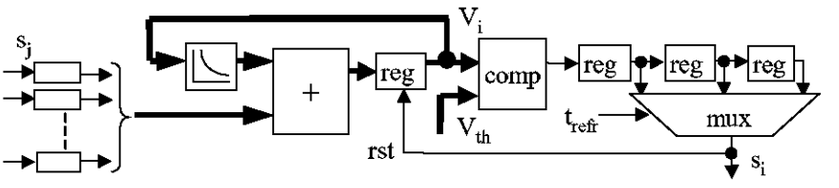


Fig. 1. Neuron block diagram

3.2 Learning Block

The learning block provides the actual value of the activation variable by measuring the time difference between spikes in the synapses between excitatory neurons.

When a spike is produced in the neuron j, the variable YDj loads its maximum value and starts to decay. Then, if the actual neuron spikes, the value of YD_j is added to the decayed value of the L variable. On the other hand, if a spike is produced first in neuron i and then in neuron j, the value of YD_i is subtracted to the decayed value of the L variable. When the L variable overcomes a certain threshold, L_{th}, positive or negative, the activation variable, A, increases or decreases respectively, unless it is already at its maximum or minimum value. If A is increased, L is reset to the value $L-2*L_{th}$, but if it is decreased, then L is reset to $L+2*L_{th}$. The diagram block of the learning block is represented in fig. 2. The YD variable has a resolution of 6 bits and the learning variable of 8 bits. The activation variable can have four values (0,1,2,3). The time constant for the variable YD is τ=20. The thresholds for the variable L are L_{th}= [-128,127] The time constant for L is 4000.

3.3 Synapse Block

The goal of the synapse block is to set the value of J (the input value added to the membrane). For each synapse a certain weight is set. This weight is multiplied by the activation variable. For this purpose, a shift register is used, so when A=0, the weight becomes 0, when A=1 the weight does not change, when A=2 the weight is multiplied by 2 and when A=3 it is multiplied by 4.

This output weight is added to the decayed value of the output J. Its decay curve depends on the type of the actual and the projecting neurons, r_i and r_j. The diagram block for the synapse block is represented in fig. 3.

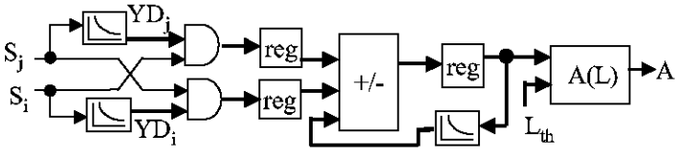


Fig. 2. Learning block

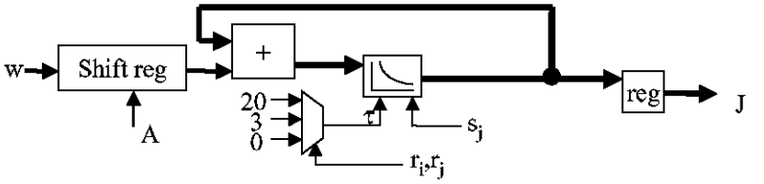


Fig. 3. Synapse block

The internal resolution of the block is 10 bits, but the output resolution is 8 bits, since the internal value of J is divided by 4 to keep the correct scaling.

The weights for an excitatory-excitatory synapse are in the range [0:32], for an excitatory-inhibitory [256:512], for inhibitory-excitatory [512:1024] and for inhibitory-inhibitory are null. The time constant τ is 20 for an excitatory-excitatory synapse, and 3 for an inhibitory-excitatory synapse. For the other two types of synapse, τ=0.

3.4 Neural Network Implementation

To test this learning strategy and the neuron structure, a neural network of size 15x15 has been used with a connectivity pattern of 24 neurons, so every cell receives spikes from a neighbourhood of 5x5.

The 20% of cells are inhibitory, and they have a random distribution in the network. The weight, w, and the initial activation variables, A, are chosen randomly.

3.4.1 Stimuli

A dynamic gradient stimuli has been applied to the neural network. As it is represented in fig.4 , a sequence of vertical bars of gradient value move through the columns of the neuron array, incrementing their frequency, in forward sense during the training, and in forward and reverse sense during the test.

The input applied to each neuron has the next parameters:

T_{CLK}: 20 ns. Maximum amplitude:127.

1. Training period: 20 us. Forward sense
2. Test period: 10 us. Forward and Reverse sense

Gaussian noise is applied to all neurons during all the time: Mean 0, s.d.= 48

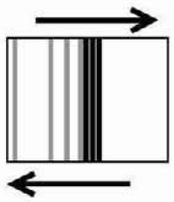
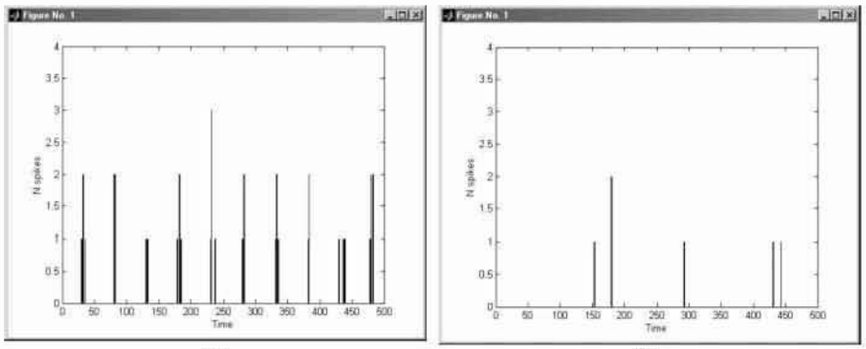


Fig. 4. Input signal applied to the neural network. The arrow to the right means forward sense and the arrow to the left means reverse sense.

3.4.2 Simulation Results

The activity calculated over a "strip" of neurons perpendicular to the direction of the movement represents a measure of "local" activity. In this case, the strip is one-column wide. In figure 5 the "local" activity is measured by the count of spikes produced as a function of the time steps. We can observe how in the forward sense there exists an activation pattern with a temporal correlation, but in reverse sense the network output has not this temporal correlation.

This result demonstrates that the selected structure of our neural network is able to perform an implicit recognition of dynamic features based on simple STDP rules.



(a) (b)

Fig. 5. Local activity in column 1 In a) when test stimuli is applied in forward sense In b) when test stimuli is applied in reverse sense

4 Hardware Simplification

Once a correct behaviour has been observed for the network, the next step consists in trying to optimise the resolution used to represent its internal parameters. This analysis will try to facilitate the hardware implementation of the model. In a first trial, the resolution of the parameters has been reduced in 2 bits Therefore, the new membrane resolution is 10 bits and the threshold is set to +160. The resolution of the output of the synapse, J, is 6 bits and the weights for an excitatory-excitatory synapse are in the range [0:32], for an excitatory-inhibitory [256:512], for inhibitory-excitatory [512:1024] and for inhibitory-inhibitory are null. And the resolution for the YD variable is 4 bits and for the L variable is 6 bits in the learning block implementation. The time constants keep being the same.

The decayment block has been modified to behave just as the algorithm explained in [3] but removing the operations with external parameters. This is because these parameters will be different for each variable, but constant during all the stimuli period. The diagram block for its implementation is presented in fig. 6.

The input signal *x* will be the input of a shift register, when the *load* signal becomes active. This shift register will divide the input by a fixed value (i.e. 8 for the J decayment), which depends on the input signal. The output of the shift register will be the value to be subtracted. The output of a counter will be compared to a maximum value. When they are equal, the counter is reset and a register loads the output of the subtraction. The output of this register will be the new value for the signal x. From a set of possible maximum times (t_1.. t_n), the value of x will choose one to establish when the subtraction has to be performed. The synapse block will only be used when the actual neuron is excitatory (i.e. r=0). The output of an inhibitory-inhibitory synapse is always

null. As well as in an excitatory-inhibitory synapse, the values of w and A are constant. Therefore since there is no memory in the synapse, the output will be w when there is a spike and 0 when there is not. For the inhibitory-excitatory synapse, the decay constant is very low so it decreases very quickly. As consequence, the synapse block can be substituted by a shift register, which divides its value at each time step by 2. In the YD decayment block, the value to subtract will be equal to 1since the YD resolution has been reduced to 4 bits,. The decayment block of the spiking neuron can be simplified, since it is possible to avoid the time control in the decayment of the membrane value. It will always decrease, because it will receive almost continuously a loaded input (i.e. noise).

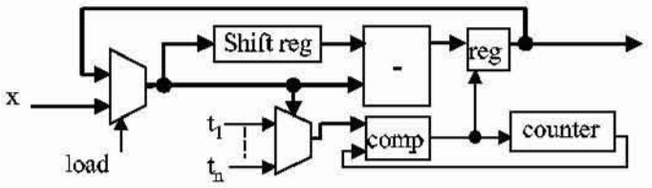


Fig. 6. Simplification of the decayment block

4.1 Simulation Results

The stimuli parameters have also been changed, but only their values, not the signal neither the training and test times. Now the maximum amplitude of the input signal will be 32 and the noise will keep its mean to 0, but its s.d. will be 12. The neighbourhood is still of 5x5, and the 20% of neurons are also inhibitory. The output of the same column is represented in fig. 7. The count of spikes produced in the same time step is represented as a function of the time steps.

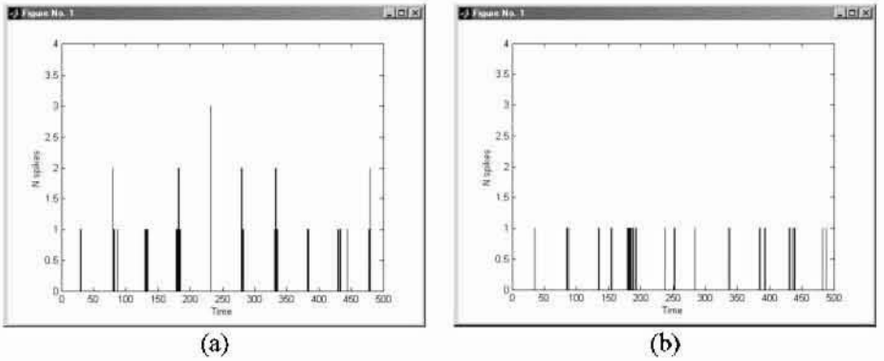


Fig. 7. Local activity in column 1 in a simplified implementation,. In a) when test stimuli is applied in forward sense. In b) when test stimuli is applied in reverse sense.

Comparing these results with those of the non-simplified implementation of section 3.4.2, we can observe how the simulation in forward sense keeps the activity pattern with a temporal correlation, but in reverse sense the activity has increased, although it has not the temporal correlation observed in forward sense, so there is still learning.

5 Conclusions

We have described the digital hardware implementation of a novel STDP rule applied to spiking neuron models and we have demonstrated that it can be used to implement learning tasks in a neural network of size 15x15. It is still possible to obtain good results with a simplified hardware description of the model, although we are on the edge of having a learning behaviour in a network of this size. Further hardware optimisation of the model is investigated by serialising the dataflow of operations corresponding to a neuron. A hardware prototype is being constructed in order to test the model using real world applications.

6 Acknowledgments

This project is funded by the grant IST-2000-28027 (POEtic) of the European Community and by grant OFES 00.0529-2 of the Swiss government.. The information provided is the sole responsibility of the authors and does not reflect the Community's opinion. The Community is not responsible for any use that might be made of data appearing in this publication.

7 References

1. Abeles, M.: Corticonics: neural circuits of the cerebral cortex, Cambridge University Press (1991).
2. Maass,W., Bishop,C.M.(eds.):Pulsed Neural Networks. The MIT Press. (2001).
3. Hill, S.L., Villa, A.E.P.: Dynamic transitions in global network activity influenced by the balance of excitation and inhibition, Network: Computation in Neural Systems 8: 165-184 (1997).
4. Fusi, S.: Long term memory: encoding and storing strategies of the brain. Neurocomputing 38-40. (2001) 1223-1228
5. Bi, G.Q., Poo, M.M.: Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength and post synaptic cell type. Journal of Neuroscience 18: 10464-10472 (1998).
6. Eriksson, J., Torres, O., Mitchell, A., Tucker, G., Lindsay, K., Halliday, D., Rosenberg, J., Moreno, J.M., Villa, A.E.P. :Spiking Neural Networks for Reconfigurable POETIC Tissue. The 5th International Conference on Evolvable Systems (2003).

Implementing a Margolus Neighborhood Cellular Automata on a FPGA

Joaquín Cerdá[1], Rafael Gadea[1], and Guillermo Paya[1]

[1] Group of Digital Systems Design, Dept. Of Electronic Engineering, Polithecnic University of Valencia,
46022 Valencia, Spain
{joacerbo, rgadea, guipava}@eln.upv.es
http://dsd.upv.es

Abstract. Margolus neighborhood is the easiest form of designing Cellular Automata Rules with features such as invertibility or particle conserving. In this paper we introduce a notation to describe completely a rule based on this neighborhood and implement it in two ways: The first corresponds to a classical RAM-based implementation, while the second, based on concurrent cells, is useful for smaller systems in which time is a critical parameter. This implementation has the feature that the evolution of all the cells in the design is performed in the same clock cycle.

1 Introduction

Cellular Automata (CA) model massively parallel computation and physical fenomena [1]. They consist of a lattice of discrete identical sites called *cells*, each one taking a value from a finite set, usually a binary set. The value of the cells evolve in discrete time steps according to deterministic rules that specify the value of each site in terms of the values of the neighboring sites. This is a parallel, synchronous, local and uniform process [1, 2, 5, 10, 14].

CA are used as computing and modeling tools in biological organization, self-reproduction, image processing and chemical reactions. Also, CA have proved themselves to be useful tools for modeling physical systems such as gases, fluid dynamics, excitable media, magnetic domains and diffusion limited aggregation [13, 3, 4, 5, 8, 12]. CA have been also applied in VLSI design in areas such as generation of pseudorandom sequences and their use in built-in self test (BIST), error-correcting codes, private-key cryptosystem, design of associative memory and testing the finite state machine [9, 10, 11].

1.1 Invertible Cellular Automata

A CA is invertible when its global function is a bijection, i.e., if every configuration – which, by definition, has exactly one successor – also has exactly one predecessor [6].

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 121-128, 2003.

Invertibility is such a desiderable property, because in the context of dynamical systems coincides with microscopic reversibility. One of the most common ways of constructing invertible cellular automata is by using a partitioning schema [6].

Partitioning Cellular Automata (PCA) are based on a different kind of local map that takes as input the contents of a region and produces as output the new state of the whole region (rather than of a single cell). This way, the state space is completely divided into non-overlapping regions. Clearly, information cannot cross a partition boundary in a single time step. In order to exchange information between regions, partitions must change at the next step.

The partitioning format is specially good for many applications because it makes very easy to construct invertible rules. If the local map is invertible then the corresponding global map is also invertible. Moreover, it is possible to construct rules wich conserve "particles", "momentum" and other desired quantities by only impose some constraints to the local map.

1.2 Margolus Neighborhood

The most important partitioning scheme is the Margolus Neighborhood, introduced in [3]. In this neighborhood each partition is 2x2 cells as shown in figure 1. We alternate between even partitions (solid lines) and odd partitions (dotted line) in order to couple them all. Periodic boundary conditions are assumed.

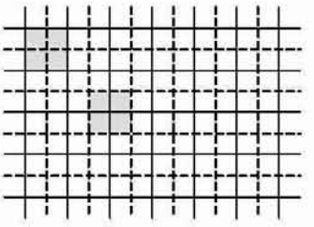


Fig. 1. Margolus Neighborhood: even (*solid lines*) and odd (*dotted lines*) partitions of a two-dimensional array into 2x2 blocks. One block in each partition is shaded

Several rules based on Margolus neighborhood have been introduced in different areas. Among them we can mention BBMCA (see figure 2) introduced by Margolus itself and capable of universal computation [3], rules TM and HPP for modelling gases [8] and the rule DIAG_then_DOWN for simulating particles that fall down and pile up [7].

We can introduce a convenient notation to assign a 'rule number" to the great quantity of rules that can be generated based on Margolus neighborhood. This notation is similar to that established by Wolfram for 1D binary CA for which the neighborhood has radius 1 [1, 2, 14]. First of all, let's assign a name to each cell in the block, as done in figure 2. Next, we can codify each transition for the rule

assigning a 0 to a white square and a 1 to a black square. The values of cells obtained from each of the 16 possible four-cells block configurations are combined to form four binary numbers that can be expressed as decimal integers. This procedure is illustrated in figure 3, that codifies the rule for BBMCA [8].

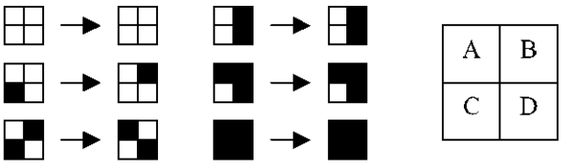


Fig. 2. BBMCA rule (*left*): this is an example of conservative rule that is invertible. The block on the right introduces a notation to refer to the cells into the partition

It's easy to see, from the tables presented in figure 3, that Margolus neighborhood allows a total number of $(2^{16})^4$ rules. From this incredibly large possibilities, only $16!$ of them are invertible, and less are particle conserving. BBMCA rule is one of these.

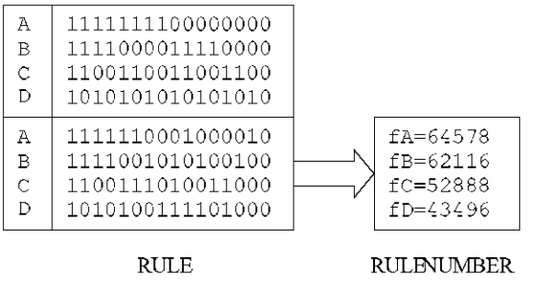


Fig. 3. Assignment of rule numbers to cellular automata based on Margolus neighborhood. The example shown is for the BBMCA rule

2 Sequential Implementation of Margolus neighborhood

A classical architecture for implementing Margolus neighbourhood at VLSI is presented in figure 4. In this approach, the lattice of cells is stored in a RAM memory disposed as a 2D array of single bits. The control path has to generate the proper signals to read the four positions that form a block and present them to the process unit. The process unit applies the transformation codified by the local map and after it is finished, the control generates the signals to store the result back in the memory.

It is necessary for the control unit to have an input from a parity generator that ensures the correct alternancy between even and odd evolutions, so the directions in the memory to be accesed are different in each case.

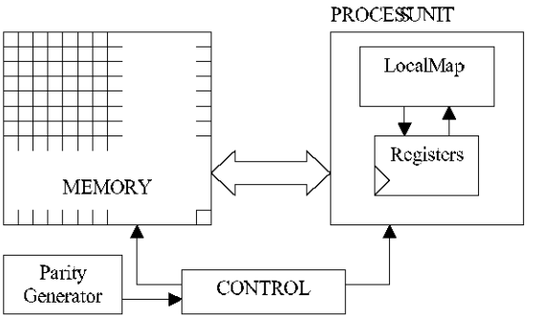


Fig. 4. Sequential implementation of Margolus neighborhood. The value of the cells is stored in a RAM array

The proposed architecture was implemented on an ALTERA FLEX10K FPGA. The logic synthesis was performed using LeonardoSpectrum Version 2002b.21 from Exemplar Logic. For the description of the circuit we used VHDL for all the components to be completely generic except for memory, that was implemented directly instantiating one LPM. The implementation and simulation was performed with ALTERA MAX+PLUSII.

The results of the synthesis were the following: the design needed about 80 Logic Cells[1] to be implemented in a FLEX10K device. The maximum size of the matrix of cells depends on the total amount of RAM embedded on the chip. Choosing a device from the family, concretely the EPF10K70RC240, we find that te total amount of memory available in Embedded Array Blocks (EABs) is 18,432 bits, that gives support to easily implement arrays of 128x128 cells[2]. After implementation the maximum operation frequency was 27.85MHz.

The main drawback of the design is the fact that the time required to perform an evolution increases linearly with the total size of the array. The control unit needs 4 clock cycles to read a 2x2 block from memory and 4 cycles to store them back. Also, the process unit needs one cycle to perform the evolution of the partition, so 9 clock cycles are necessary to evolve each 2x2 block from memory. All this information is summarized in table 1.

---

[1] In fact, it presents a little variation (from 70 to 90 LCs) depending on the total size of the matrix of cells.

[2] Actually the total dimension, if we could use all the memory available, should be a little more, but the design has been thought to have size of $2^n x 2^n$.

Table 1. Timing characteristics of the sequential implementation of Margolus Neighborhood. Numbers were obtained by using a 25MHz clock frequency

Size	Total Memory	Cycles to evolve	Evolutions per second
8x8	64 bits	144	$1{,}7 \cdot 10^5$
16x16	256 bits	576	$4{,}3 \cdot 10^4$
32x32	1024 bits	2304	$1{,}1 \cdot 10^4$
64x64	4096 bits	9216	$2{,}7 \cdot 10^3$
128x128	16384 bits	36864	$6{,}8 \cdot 10^2$

Most of this drawbacks can be overtaken by relying on more powerful families, using the special features given by the manufacturers. For intance, if we choose to use an ALTERA APEX 20K FPGA we can get advance of the embedded dual-port RAM that allows the user to perform simultaneous read-and-write operations. This can reduce the number of cycles needed to perform a block evolution from 9 to 5, almost doubling the number of evolutions per second that the system can achieve.

3. Concurrent Implementation

In some applications, evolution rates like those presented in the previous paragraph are non acceptable. The main advance of a Celular Automata Structure is precisely its high paralelisation degree, and the implementation proposed converts it into a serial scheme to perform the matrix actualisation. If we want to obtain a real concurrent Cellular Automata, new strategies need to be explored.

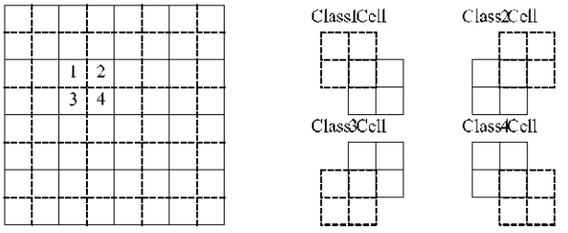


Fig. 5. Classes of cells in Margolus neighbourhood. Each class of cells behaves as one type of cell of those introduced in figure number 2 different for each even/odd evolution

3.1 Proposed Architecture

If we carefully study the connection schema of a Margolus Neigborhood Cellular Automata, easily we can distinguish between four classes of cells, depending on its

position into the global matrix. In figure 5 these four classes are shown. Class 1 Cells are updated in odd cycles as Type D cells (referred to the notation introduced in figure 2) from a dotted line block, and are updated in even cycles as Type A Cells from a solid line block. The same way, class 2 Cells are updated in odd cycles as Type C cells from a dotted line block, and are updated in even cycles as Type B cells from a solid line block. The rest of classes and their connectivities are easily inferred from the figure.

This connection structure leads us to reduce all classes of Cells to a common structure that is depicted on figure 6. For each cell in the matrix we need to define two different functions: the even one and the odd one. A multiplexer selects between results on even or odd branches depending on the present cycle. Also, to easily supply initial data to the circuit, we have included a second multiplexer for data sinchronous load. Finally, an Enable terminal has been added to hold and start the evolution.

If we fix the proposed architecture for all the classes of cells in the matrix, the only difference between classes is the way the neighbor cells are connected to the inputs of the even/odd functions. Table 2 summarizes these connections for each class of cells.

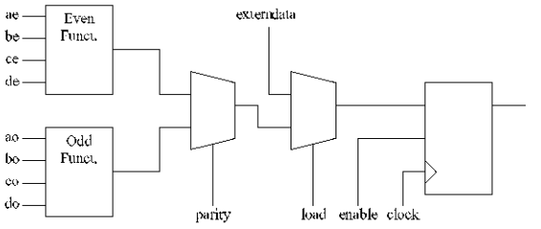


Fig. 6. Common structure of a cell. Each one supports two different functions: the even one and the odd one

Table 2. Connectivities for all four classes of cells in Margolus neighborhood. Names of inputs are referred to the terminology stated in figure 6

Cell Class	ae	be	ce	de	ao	bo	co	do
1	itself	right	down	down-right	up-left	up	left	itself
2	left	itself	down-left	down	up	up-right	itself	right
3	up	up-right	itself	right	right	itself	down-right	down
4	up-left	up	left	itself	itself	right	down	down-right

3.2 Implementation and Results

This concurrent architecture was implemented on an ALTERA FLEX10K FPGA. The logic synthesis was performed using LeonardoSpectrum Version 2002b.21 from Exemplar Logic. Due to the fact that no memory was needed for implementing this design, VHDL could be used for the complete description, giving a complete device independence and allowing us to sinthesize on different fanilies, such as some from XILINX. The implementation and simulation was performed with ALTERA MAX+PLUSII.

FLEX10K Logic Cell contains a four-input look-up-table that is specially indicated for implementing one of the even/odd functions. So, the complete Cell showed in figure 6 needs 4 logic cells to be implemented: two for the two functions and two for the multiplexers and register. Only 9 additional LCs are needed to generate the control signals for the main design.

Results of the implementations for a 8x8 matrix are the folowing: Circuit needs a total of 265 LC to be implemented on a EPF10K70RC240. This means only a 7% of the total LCs included in the device. In fact, the device containt 3744 LCs, that are enough to support matrix of size 30x30[3]. The maximum frequency is 63.69MHz, that is clearly greater than that obtained in the sequential implementation. But the main advantage accomplished is the way the circuit evolves: it updates the whole matrix in the same clock cycle, independently of the size. So, for any size of the array, the rate of evolutions per second is constant and equal to one clock period.

It is important to remark that different FPGA structures can be evaluated for giving support to the presented implementation, some more appropriated to the structure shown. If we choose a XC4000 FPGA from XILINX we see that a CLB incorporates 2 LUTs, reducing the number of CLB per cell to 2. However, it is known that an XC4000 CLB is 2,375 times greater than a FLEX10K LC, so we selected a FLEX10K device for our implementations.

3.3 Comparison between implementations

The two main differences between both strategies are evident from the given results: sequential implementation permits large matrix sizes, thus having the drawback of that the time per evolution increases linearly with the size of the matrix. On the other hand, concurrent implementation is more adequated when time is a critical parameter, even though the sizes obtained are small. This could be indicated in some VLSI applications such as random number generation or BIST.

---

[3] In this case, design is not limited to sizes of 2^nx2^n. With this implementation the only limitation, characteristical of Margolus neighborhood, is that size must be the square of an even number.

4. Conclusions

We have studied Margolus neighborhood Cellular Automata both its theoretical fountations and its applications. We have proposed a notation to give a number to specify completely a rule. This rule is easy to be constructed invertible or particle conserving.

We propose two implementations for Margolus neighborhood Cellular Automata, one RAM based that allows large sizes but offers poor timing characteristics as processing times increases linearly with memory size. Other implementation, that we call concurrent, gives support to much small systems in which time is a critical factor, performing a complete evolution in only one clock cycle.

References

1. Wolfram, S.: Cellular Automata. Los Alamos Science, 9 (1983) 2–21
2. Wolfram, S.: Statistical mechanics of cellular automata. Reviews of Modern Physics, 55 (1983) 601–644
3. Margolus, N.: Physics-Like models of computation. Physica 10D (1984) 81-95
4. Toffoli, T.: Cellular Automata as an alternative to (rather tah an approximation of) Differential Equations in Modeling Physics. Physica 10D (1984) 117-127
5. Toffoli, T.: Occam, Turing, von Neumann, Jaynes: How much can you get for how little? (A conceptual introduction to cellular automata). The Interjournal (October 1994)
6. Toffoli, T., Margolus, N.: Invertible cellular automata: a review. Physica D, Nonlinear Phenomena, 45 (1990) 1–3
7. Gruau, F. C., Tromp, J. T.: Cellular Gravity. Parallel Processing Letters, Vol. 10, No. 4 (2000) 383–393
8. Smith, M. A.: Cellular Automata Methods in Mathematical Physics. Ph.D. Thesis. MIT Department of Physics (May 1994).
9. Wolfram, S.: Cryptography with Cellular Automata. Advances in Cryptology: Crypto '85 Proceedings, Lecture Notes in Computer Science, 218 (Springer-Verlag, 1986) 429–432
10. Sarkar, P.: A brief history of cellular automata. ACM Computing Surveys, Vol. 32, Issue 1 (2000) 80–107
11. Shackleford, B., Tanaka, M., Carter, R.J., Snider, G.: FPGA Implementation of Neighborhood-of-Four Cellular Automata Random Number Generators. Proceedings of FPGA 2002 (2002) 106–112
12. Vichniac, G. Y.: Simulating Physics with Cellular Automata. Physica 10D (1984) 96-116
13. Popovici, A. , Popovici, D.: Celluar Automata in Image Processing. Proceedings of MTNS 2002 (2002)
14. Wolfram, S.: A New Kind of Science. Wolfram media (2002)

An Empirical Comparison of Training Algorithms for Radial Basis Functions[1]

Mamen Ortiz-Gómez, Carlos Hernández-Espinosa,
and Mercedes Fernández-Redondo.

Universidad Jaume I, Campus de Riu Sec, D. de Ingeniería y Ciencia de los Computadores,
12071 Castellón, Spain
{mortiz,espinosa,redondo}@icc.uji.es

Abstract. In this paper we present a review and comparison of five different algorithms for training a RBF network. The algorithms are compared using nine databases. Our results show that the simplest algorithm, k-means clustering, may be the best alternative. The results of RBF are also compared with the results of Multilayer Feedforward with Backpropagation, the performance of a RBF network trained with k-means clustering is slightly better and the computational cost considerably lower. So we think that RBF may be a better alternative.

1 Introduction

Perhaps, Multilayer Feedforward and Radial Basis Functions are the two most employed neural networks in applications.

Comparing both neural networks, Radial Basis Functions (RBF) has the advantage of fast training. Other aspect like the relative generalization capability and other properties are not yet clear in the bibliography.

A RBF has two layer of networks (without considering the input units). The first layer is composed of neurons with a Gaussian transfer function and the second layer (the output units) has neurons with a linear transfer function. The output of a RBF network can be calculated with equation 1.

$$F_k(x) = \sum_{q=1}^{Q} w_q^k \cdot \exp\left(\sum_{n=1}^{N} \left(C_{q,n}^k - X_n\right)^2 \Big/ \left(\sigma_q^k\right)^2 \right) \quad (1)$$

Where ${C_{q,n}}^k$ are the components of the center of the Gaussian functions, σ_q^k control the width of the Gaussian functions and ${w_q}^k$ are the weights among the Gaussian units and the output units.

[1] This work was supported by a Proyect of Generalitat Valenciana number GV01-14

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 129-136, 2003.

As the equation 1 shows, there are three elements to design in the neural network, the center and widths of the Gaussian units and the weights among the Gaussian units and output units.

There are two different procedures to design the network. One is to train simultaneously the centers, widths and weights in a full supervised fashion [1,2], similar to the algorithm Backpropagation for Multilayer Feedforward. This procedure has the same drawbacks of Backpropagation, long training time and high computational cost.

The second is to train the networks in two steps. First we find the centers and widths by using same unsupervised clustering algorithm and after that we train the weights among hidden units and output units by a supervised algorithm. This process is usually fast, and the most important step is the training of centers and widths because for the weights the relation is linear and simple procedures like LMS can be used.

Therefore, in this paper we will focus on this last type of unsupervised-supervised algorithms.

In the bibliography there are many of these algorithms [3-7] and it is not clear their relative performance because of a lack of comparison.

So in this paper we present a comparison of five of these algorithms.

The organization of the paper is as follows. In section two we briefly review the training algorithms which are compared. In section three we present the experimental results and we finally conclude in section four.

2 Theory

In this section, we briefly review the training algorithms which are compared.

2.1 Algorithm 1

This training algorithm is the simplest one. It was proposed in [3]. It uses adaptative k-means clustering to find the centers of the Gaussian units. We successively present an input pattern and after each presentation, we find the closest center and adapt the center towards the input pattern, according to equation 2.

$$c(n+1) = c(n) + \eta \cdot (x - c(n)) \tag{2}$$

Where x is the input pattern, c the closest center and η the adaptation step.

After finding the centers, we should calculate the widths of the Gaussian units. For that, it is used a simple heuristic, we calculate the mean distance between one center and one of the closets neighbors, P, for example, the first ($P=1$), second ($P=2$), third ($P=3$) or fourth ($P=4$) closest neighbor.

We need a trial and error procedure to design the network because the number of centers should be fixed a priori. In our experiments we have tried 10, 20, 30, 40 ,50, 60, 70 , 80, 90, 100 and 110 centers and for the widths we have used P=1, 2, 3 and 4.

2.2 Algorithm 2

It was proposed in reference [4]. The clustering algorithm is the following:

1. Initially assign each training point to a unique cluster ($K=1, \ldots, C$)
2. Select the first cluster.
3. Find any clusters of the same class.
4. Merge the two clusters and compute the new mean.
5. Compute the distance, d_{opp}, from the new mean to the mean of the nearest cluster of the opposite class.
6. Compute the distance from the new mean to the furthest point in its new cluster, which we shall define as being the radius, R, of the cluster.
7. If $d_{opp}>\alpha \cdot R$ (α is a constant), then accept the merge in 4, and start again from step 3, associating the current value of K with the newly created cluster, and setting $C=C-1$. If this condition is not satisfied, reject the merge and recover the two original clusters, then restart from step 3, but K and C remains unchanged. Repeat steps 3 through 7 until all clusters are considered and then change $K=K+1$.
8. Repeat steps 3 through 7, until K is finally equal to C.

After finding the centers by the algorithm, the widths are calculated. But in this case, it is used a matrix Σ instead a single value for the widths. The equation of the neural network for this case is in equation 3.

$$F_k(x) = \sum_{q=1}^{Q} w_q^k \cdot \exp\left(\left(x - c_q\right)^T \cdot \sum_q^{-1} \cdot \left(x - c_q\right)\right) \tag{3}$$

The procedure for calculating these matrices is complex and should be looked for in the reference.

2.3 Algorithm 3

This algorithm is proposed in reference [5]. However, we have slightly modified the algorithm, in the original reference it is used a truncation for the Gaussian units and non-RBF functions in the hidden layer. We have applied the algorithm without truncation in the Gaussian units and with only RBF unit in the hidden layer.

Basically the algorithm is the following. The Gaussian units are generated incrementally, in stages, by random clustering. Let k ($=1,2,3, \ldots$) denote a stage of this process. A stage is characterized by a parameter δ that specifies the maximum radius for the hypersphere that includes the random cluster of points that is to define the Gaussian unit, this parameter is successively reduced in every stage k ($\delta_k=\alpha \cdot \delta_{k-1}$, with α in the range 0.5-0.8). The Gaussian units at any stage k are randomly selected in the following way. Randomly select an input vector x_i from the training set and search for all other training vectors within the δ_k neighborhood of x_i. The training vector are used to define the Gaussian unit (the mean is the center, and the standard deviation the width) and then removed from the training set. To define the next Gaussian unit another input vector x_i is randomly selected and the process repeated. This process of

randomly picking an input vector x_i is repeated until the remaining training set is empty. Furthermore, when the number of points in the cluster is less than a certain parameter β no Gaussian unit is created. The stages are repeated until the cross-validation error increases.

The algorithm is complex and the full description can be found in the reference.

2.4 Algorithm 4

It is proposed in reference [6].They use a one pass algorithm called APC-III, clustering the patterns class by class instead of the entire patterns at the same time. The APC-III algorithm uses a constant radius to create the clusters, in the reference this radius is calculated as the mean minimum distance between training pattern multiplied by a constant α, see equation 4.

$$R_0 = \alpha \cdot \frac{1}{P} \cdot \sum_{i=1}^{P} \min_{i \neq j} \left(\left\| x_i - x_j \right\| \right) \tag{4}$$

The algorithm is basically the following:

1. Select on input pattern and construct the first cluster with center equal to this pattern
2. Repeat steps 3 to 5 for each pattern
3. Repeat step 4 for each cluster
4. If the distance between the pattern and the clusters is less than R_0 include the pattern in the cluster and recalculate the new center of the cluster. Exit the loop 3.
5. If the pattern is not included in any cluster then create a new cluster with center in this pattern.

The widths are calculated with the following heuristic: find the distance of the center to the nearest cluster which belongs to a different class and assign this value multiplied by β to the width.

2.5 Algorithm 5

This algorithm is proposed in reference [7]. However, we have slightly modified the algorithm, in the original reference it is used a truncation for the Gaussian units and a hard limiting function for the output layer. We have applied the algorithm without these modifications of the normal RBF network.

The description of the algorithm is as follows. The Gaussian units are generated class by class, so the process is repeated for each class. In a similar way to algorithm 3 the Gaussian units are generated in stages. A stage is characterized by its majority criterion, a majority criterion of 60% implies that the cluster of the Gaussian unit must have at least 60% of the patterns belonging to its class. The method will have a maximum of six stages, we begin with a majority criterion of 50% and end with 100%, by increasing 10% in each stage. The Gaussian units for a given class k at any stage h are randomly selected in the following way. Randomly pick a pattern vector x_i of class

k from the training set and expand the radius of the cluster until the percentage of patterns belonging to the class falls below the majority criterion, then the patterns of class k are used to define the Gaussian unit (the mean is the center and the standard deviation is the width) and are removed from the training set. When the number of pattern in the cluster is below than a parameter, β, no Gaussian unit is created. To define the next Gaussian unit another pattern x_i of class k is randomly picked from the remaining training set and the process repeated. The successive stage process is repeated until the cross-validation error increases.

3 Experimental Results

We have applied the five training algorithms to nine different classification problems. They are from the UCI repository of machine learning databases. Their names are Balance Scale (BALANCE), Cylinders Bands (BANDS), Liver Disorders (BUPA), Credit Approval (CREDIT), Glass Identification (GLASS), Heart Disease (HEART), the Monk's Problems (MONK'1, MONK'2) and Voting Records (VOTE). The complete data and a full description can be found in the UCI repository (http://www.ics.uci.edu/~mlearn/MLRepository.html).

The first step was to determine the appropriate parameters of the algorithms by trial and error. For the algorithm 1 we have to fix the number of Gaussian units, N, and the parameter P. We have tried all combinations of P=1, 2, 3, 4 and N=10, 20, 30, …, 110. For the algorithm 2 we have to choose the value of parameter α, we have tested the values 1, 1.25, 1.5, 1.75, 2, 2.25, 2.5, 2.75 and 3. In the algorithm 3 we have tested all the combination of parameter β=3, 5, 8 and parameter α=0.5, 0.65, 0.8. For the algorithm 4 we have used all the combinations of parameters β=5, 6, 7, 8 and parameter α=1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7. In the algorithm 5 we have to choose the parameter β, we have tested the values 3, 5, 8. The final selected values of the parameters obtained by cross-validation for each database are in table 1.

As we can see from the results of the parameters, algorithm 3, 4 and 5 are rather insensitive to the parameters.

After that, with the final parameters we trained ten networks with different partition of the data in training, cross-validation and test, also with different random parameters. With this procedure we can obtain a mean performance in the database (the mean of the ten networks) and an error in the performance calculated by standard error theory.

These results are in Table 2. We have for each database the mean Percentage in the test and the mean number of clusters, i. e., the number of Gaussian units in the network.

Table 1. Selected values of the parameters

		Balance	Band	Bupa	Credit	Glass	Heart	Monk1	Monk2	Vote
Alg. 1	Clusters	30	60	10	20	100	100	90	90	40
	P	4	2	3	2	1	1	2	2	4
Alg. 2	α	2.25	1.25	1	2	2	1	2.25	2	1
Alg. 3	β	5	3	5	3	5	3	3	8	3
	α	0.8	0.65	0.8	0.8	0.8	0.8	0.8	0.8	0.65
Alg. 4	β	5	7	5	6	5	5	5	5	5
	α	1.7	1.3	1.3	1.7	1.2	1.7	1.7	1.4	1.7
Alg. 5	β	5	3	3	3	3	3	3	3	5

Table 2. Performance of the different algorithms, Radial Basis Functions.

	TRAINING ALGORITHM							
	Alg. 1		Alg. 2		Alg. 3		Alg. 4	
DATABASE	Perc.	Cluster	Perc.	Cluster	Perc.	Cluster	Perc.	Cluster
Balance	88.5±0.8	30±0	68±6	395±0	87.6±0.9	88.5±1.6	88.0±0.9	94.7±0.5
Band	74.0±1.5	60±0	61±3	49±3	67±2	18.7±1.0	67±4	97.2±0.3
Bupa	59.1±1.7	10±0	54.3±1.9	17.8±0.3	57.6±1.9	10.3±1.5	60±4	106.2±0.3
Credit	87.3±0.7	20±0	83.7±0.8	418±0	87.5±0.6	95±14	87.9±0.6	161.10±0.17
Glass	89.6±1.9	100±0	78.2±1.2	111.5±1.3	79±2	30±2	82.8±1.5	59.9±0.7
Heart	80.8±1.5	100±0	63±3	12.6±0.7	80.2±1.5	26±4	72±4	71.8±0.6
Monk1	76.9±1.3	90±0	67±4	282±0	72±2	93±8	68±3	97.4±0.6
Monk2	71.0±1.5	90±0	73±2	282±0	66.4±1.7	26±4	66.5±0.8	143±0
Vote	95.1±0.6	40±0	61±5	11.9±0.7	93.6±0.9	53±5	94.1±0.8	120.30±0.15

Table 2. Continuation

	TRA. ALG.	
	Alg. 5	
DATABASE	Perc.	Cluster
Balance	87.4±0.9	45±7
Band	65.8±1.4	4.5±1.3
Bupa	47±3	11±5
Credit	86.4±0.9	32±4
Glass	81.2±1.8	22±2
Heart	78±3	10±2
Monk1	64±2	23±6
Monk2	71.6±1.5	20±2
Vote	76±5	5.0±1.1

The results of Table 2 show that the best training algorithm is the simplest, number 1. But it has the drawback of a great number of trials for the selection of parameters, we should probe all the combinations of P and number of clusters and select the appropriate combination by trial and error.

It is curious that the simplest algorithm provides the best generalization. Specially in the case that algorithms 2, 3, 4 and 5 were proposed after algorithm 1. The explanation is in the following paragraphs.

The performance of algorithm 2 is rather low compared with algorithm 1. In the original reference it was tested in only two classification problems. One artificial two dimensional problem of two Gaussian distributions with overlapping and one artificial eight-dimensional problem also of Gaussian distributions. The algorithm was compared with what they call a standard RBF, but they build the RBF with a number of Gaussian units equal to the number of training patterns. The expected generalization of this procedure should be low.

In the results of algorithm 3 in the original reference. The test set was used at the same time as cross-validation set. This is a non-standard procedure that can falsify the results, in fact, in our experiments the cross-validation results are clearly better than the test results. Furthermore, the algorithm was not compared with anyone.

The algorithm 4, was tested in only one character recognition problem and it was compared with the k-means clustering in the original reference. But they do not optimize the number of clusters of k-means clustering by trial an error. They just used the same value obtained by APC-III (15 clusters per class) and this procedure is not optimal for k-means clustering.

Finally, algorithm 5 was tested in the original reference in four artificial problems of Gaussian distributions and four real problems, one of them heart. For comparison we reproduced the results of the database heart. The k-means clustering got a percentage of 63.64% clearly our results (80.8±1.5) is better, it seems we have tuned the parameters better. The algorithm 5 got a percentage of 81.82% which nearly similar to our results (78±3, taking in consideration the error).

In order to perform a further comparison, we include the results of Multilayer Feedforward with Backpropagaion in Table 3.

Table 3. Performance of Multilayer Feedforward with Backpropagation

DATABASE	Number of Hidden	Percentage
BALANCE	20	87.6±0.6
BANDS	23	72.4±1.0
BUPA	11	58.3±0.6
CREDIT	15	85.6±0.5
GLASS	3	78.5±0.9
HEART	2	82.0±0.9
MONK'1	6	74.3±1.1
MONK'2	20	65.9±0.5
VOTING	1	95.0±0.4

We can see that the results of RBF with k-means clustering are slightly better. This is the case in databases Balance, Bands, Bupa, Credit Glass, Monk1, Monk2 and Voting. The only exception is database Heart where the result of Multilayer Feedforward is slightly better. The most important difference is in database Glass, where RBF obtains 89.6±1.9 and Multilayer Feedforward 78.5±0.9. Furthermore, the computational cost is considerably lower. According to our results, RBF is a better alternative than Multilayer Feedforward with Backpropagation.

4 Conclusions

In this paper we have presented a review of five training algorithms for Radial Basis Functions. The algorithms are compared using nine databases. Our results show that the simplest algorithm, k-means clustering, may be the best alternative. The results of RBF are also compared with the results of Multilayer Feedforward with Backpropagation, the performance of a RBF network trained with k-means clustering is slightly better and the computational cost considerably lower. So we think it is a better alternative.

References

[1] McLoone, S., Brown, M., Irwin, G., Lightbody, G., "A Hybrid Linear/Nonlinear Training Algorithm for Feedforward Neural Networks", IEEE Trans. on Neural Networks, vol. 9, no. 4 pp. 669-684, 1998.

[2] Krayiannis, N., "Reformulated Radial Basis Neural Networks Trained by Gradient Descent", IEEE Trans. on Neural Networks, vol. 10, no. 3, pp. 657-671, 1999.

[3] Moody, J., Darken, C.J., "Fast Learning in Networks of Locally-Tuned Procesing Units", Neural Computation, vol. 1, pp. 281-294, 1989.

[4] Musavi, M.T., Ahmed, W., Chan, K.H., Faris, K.B., Hummels, D.M., "On the Training of Radial Basis Function Classifiers", Neural Networks, vol. 5, pp. 595-603, 1992.

[5] Roy, A., Govil, S., et alt. "A Neural-Network Learning Theory and Polynomial Time RBF Algorithm", IEEE Trans. on Neural Networks, vol. 8, no. 6, pp. 1301-1313, 1997.

[6] Hwang, Y., Bang, S., "An Efficient Method to Construct a Radial Basis Function Neural Network Classifier", Neural Network, vol. 10, no. 8, pp. 1495-1503, 1997.

[7] Roy, A., Govil, S., et alt., "An Algorithm to Generate Radial Basis Function (RBF)-Like Nets for Classification Problems", Neural Networks, vol. 8, no. 2, pp. 179-201, 1995.

Ensemble Methods for Multilayer Feedforward: An Experimental Study

Carlos Hernández-Espinosa,
Mercedes Fernández-Redondo, and Mamen Ortiz-Gómez

Universidad Jaume I, Campus de Riu Sec
D. de Ingeniería y Ciencia de los Computadores, 12071 Castellón, Spain
redondo@icc.uji.es
espinosa@icc.uji.es

Abstract. Training an ensemble of networks is an interesting way to improve the performance with respect to a single network. However there are several methods to construct the ensemble and there are no complete results showing which one could be the most appropriate. In this paper we present a comparison of eleven different methods. We have trained ensembles of a reduced number of networks (3 and 9) because in this case the computational cost is not high and the method is suitable for applications. The results show that the improvement in performance from three to nine networks is marginal. Also, the best method is called "Decorrelated" and uses a penalty term in the usual Backpropagation function to decorrelate the network outputs in the ensemble.

1 Introduction

Perhaps, the most important property of a neural network is the generalization capability. The ability to correctly respond to inputs which were not used in the training set.

One technique to increase the generalization capability with respect to a single neural network consist on training an ensemble of neural network, i.e., to train a set of neural networks with different weight initialization or properties and combine the outputs of the different networks in a suitable manner to give a single output.

It is clear form the bibliography that this procedure increases the generalization capability. The error of a neural network can be decomposed into a bias and a variance [1,2]. The use of an ensemble usually keeps the bias constant and reduce the variance if the errors of the different networks are uncorrelated or negatively correlated. Therefore, it increases the generalization performance. The two key factors to design an ensemble are how to train the individual networks to get uncorrelated errors and how to combine the different outputs of the networks to give a single output.

Among the methods of combining the outputs, the two most popular are voting and output averaging [3]. In this paper we will normally use output averaging because it

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 137-144, 2003.

has no problems of ties. Furthermore, in a previous comparative study it is shown that averaging provides a reasonable performance [4].

In the other aspect, nowadays, there are several different methods in the bibliography to train the individual networks and construct the ensemble [1-3], [5-10].

However, there is a lack of comparison among the different methods and it is not clear which one can provide better results. We have only found a comparison in the bibliography for the case of neural networks as classifiers [11]. This comparison is performed on a large quantity of problems (23) but only four ensemble methods are compared: simple ensemble, bagging, adaboost and arcing.

In contrast, in this paper, we present a comparison among eleven different methods of constructing the ensemble in nine different databases.

The organization of the paper is the following. In section two we briefly describe the different methods, in section three, we present the experimental results of the comparison and finally we conclude in section four.

2 Theory

In this section we briefly review the different ensemble methods, a full description can be found in the references.

2.1 Simple Ensemble

A simple ensemble can be constructed by training different networks with the same training set but with different random initialization in the weights. In this ensemble technique, we expect that the networks will converge to different local minimum and the errors will be uncorrelated.

2.2 Bagging

This ensemble method is described in [5]. The ensemble method consists on generating different datasets drawn at random with replacement from the original training set. After that, we train the different networks in the ensemble with these different datasets (one network per dataset). We have used datasets which have a number of training points equal to twice the number of points of the original training set, as it is recommended in the reference [1].

2.3 Bagging with Noise (Bagnoise)

It was proposed in [2]. It is a modification of Bagging, we use in this case datasets of size $10 \cdot N$ (number of training points) generated in the same way of Bagging, where N is the number of training points of the initial training set. And we introduce a random noise in every selected training point drawn from a normal distribution with a small variance.

2.4 Boosting

This ensemble method is reviewed in [3]. It is conceived for a ensemble of only three networks. It trains the three network of the ensemble with different training sets. The first network is trained with the whole training set, N input patterns. After this training, we pass all N patterns through the first network, using a subset of them, such that the new training set has 50% of patterns incorrectly classified by the first network and 50% classified correctly. With this new training set we train the second network. After the second machine is trained, the N original patterns are presented to both networks. If the two networks disagree in the classification, we add the training pattern to the third training set. Otherwise we discard the pattern. With this third training set we train the third network.

In the original theoretical derivation of the algorithm, evaluation of the test performance was as follows: present a test pattern to the three networks. If the first two networks aggre, use this label, otherwise use the label assigned by the third network. In addition to this voting scheme, we have use simple averaging of the outputs in our experiments.

2.5 CVC

It is reviewed in [1]. In k-fold cross-validation, the training set is divided into k subsets. Then, k-1 subsets are used to train the network and results are tested on the subset that was left out. Similarly, by changing the subset that is left out of the training process, one can construct k classifiers, which are trained on a slightly different training set. This is the technique used in this method.

2.6 Adaboost

We have implemented the algorithm denominated "Adaboost.M1" in the reference [6]. In the algorithm the successive networks are trained with a training set selected at random from the original training set, but the probability of selecting a pattern changes depending on the correct classification of the pattern and on the performance of the last trained network. The algorithm is complex and the full description should be looked for in the reference. The method of combining the outputs of the networks is also particular to this algorithm. We have use this method and the usual output averaging in our experiments.

2.7 Decorrelated (Deco)

This ensemble method was proposed in [7]. It consists on introducing a penalty term added to the usual Backpropagation error function. The penalty term for network number *j* in the ensemble is in equation 1.

$$Penalty = \lambda \cdot d(i,j)(y - f_i) \cdot (y - f_j) \tag{1}$$

Where λ determines the strength of the penalty term and should be found by trial and error, y is the target of the training pattern and f_i and f_j are the outputs of networks number i and j in the ensemble. The term $d(i,j)$ is in equation 2.

$$d(i,j) = \begin{cases} 1, & if\ i = j-1 \\ 0, & otherwise \end{cases} \tag{2}$$

2.8 Decorrelated2 (Deco2)

It was proposed also in reference [7]. It is basically the same method but with a different term $d(i,j)$ in the penalty. In this case the expression of $d(i,j)$ is in equation 3.

$$d(i,j) = \begin{cases} 1, & if\ i = j-1 \quad and \quad i\ is\ even \\ 0, & otherwise \end{cases} \tag{3}$$

2.9 Evol

This ensemble method was proposed in [8]. In each iteration (presentation of a training pattern), it is calculated the output of the ensemble for the input pattern by voting. If the output is correctly classified we continue with the next iteration and pattern. Otherwise, the network with an erroneous output and lower MSE (Mean Squared Error) is trained in this pattern until the output of the network is correct. This procedure is repeated for several networks until the vote of the ensemble correctly classifies the pattern. For a full description of the method see the reference.

2.10 Cels

It was proposed in [9]. This method also uses a penalty term added to the usual Backpropagation error function to decorrelate the output of the networks in the ensemble. In this case the penalty term for network number i is in equation 4.

$$Penalty = \lambda \cdot (f_i - y) \cdot \sum_{j \neq i} (f_j - y) \tag{4}$$

Where y is the target of the input pattern and f_i and f_j the outputs of networks number i an j for this pattern.

The authors propose in the paper the winner-take-all procedure to combine the outputs of the individual networks, i. e. the highest output is the output of the ensemble. We have used this procedure and the usual output averaging.

2.11 Ola

This ensemble method was proposed in [10]. In this method, first, several datasets are generated by using bagging, with a number of training patterns in each dataset equal to the original number of the training set. Every network is trained in one of this datasets and in what it is called virtual data. The virtual data for network i is generated by

selecting randomly samples for the original training set and perturbing the sample with a random noise drawn from a normal distribution with small variance. The target for this new virtual sample is calculated by the output of the ensemble for this sample without network number *i*. For a full description of the procedure see the reference.

3 Experimental Results

We have applied the eleven ensemble methods to nine different classification problems. They are from the UCI repository of machine learning databases. Their names are Balance Scale (BALANCE), Cylinders Bands (BANDS), Liver Disorders (BUPA), Credit Approval (CREDIT), Glass Identification (GLASS), Heart Disease (HEART), the Monk's Problems (MONK'1, MONK'2) and Voting Records (VOTE). The complete data and a full description can be found in the UCI repository (http://www.ics.uci.edu/~mlearn/MLRepository.html).

We have constructed ensembles of a low number of networks, in particular 3 and 9 networks. We think that the ensemble methodology can be useful for an application if the number of networks in the ensemble is low. Otherwise the computational cost (of an ensemble with a high number of networks) would be high an the method impractical.

The first step was to determine the right parameters for each database, in the case of methods Cels (parameter lambda of the penalty), Ola (standard deviation of the noise), Deco and Deco2 (parameter lambda of the penalty) and BagNoise (standard deviation of the noise). The value of the final parameters obtained by trial and error is in Table 1.

With these parameters we trained the ensembles of three and nine networks. We repeated this process of training an ensemble ten times for different partitions of data in training, cross-validation and test sets. In order to obtain a mean performance of the ensemble for each database (the mean of the ten ensembles) and an error in the performance calculated by standard error theory. The results of the performance are in table 2 for the case of ensembles of three networks and in table 3 for the case of nine.

Table 1. Parameters for different ensemble methods

	Cels		**Ola**		**Deco**	**Deco2**	**Bag Noise**
	Networks in Ensemble		Networks in Ensemble				
	3	9	3	9			
Balance	0.1	0.1	0.5	0.5	1	0.6	0.1
Band	0.5	0.25	0.5	0.5	1	0.6	0.2
Bupa	0.75	0.1	0.5	0.6	0.8	0.2	0.1
Credit	0.1	0.75	0.6	0.6	0.2	0.2	0.7
Glass	0.25	0.1	0.3	0.2	0.6	0.6	0.2
Hear	0.5	0.25	0.3	0.4	0.6	0.6	0.4
Monk1	0.25	0.1	0.2	0.2	0.6	0.6	0.1
Monk2	0.1	0.1	0.6	0.6	0.4	0.8	0.1
Vote	0.5	0.75	0.3	0.3	0.4	0.6	0.1

Table 2. Performances of the different methods for an ensemble with three networks

	DATABASE								
METHOD	**Balance**	**Band**	**Bupa**	**Credit**	**Glass**	**Hear**	**Monk1**	**Monk2**	**Vote**
Single Network	87.6±0.6	72.4±1.0	58.3±0.6	85.6±0.5	78.5±0.1	82.0±0.9	74.3±1.1	65.9±0.5	95.0±0.4
Adaboost	95.9±0.5	73.1±1.4	72.4±1.7	85.8±1.0	92.8±1.6	81±2	70±7	79.0±1.8	95.6±0.7
Bagging	94.6±0.9	72.9±1.3	72.6±1.6	87.2±0.5	93.8±1.1	84.2±1.1	98.3±1.0	87±1.6	96.1±0.7
Bag Noise	90.6±0.7	76.2±1.0	64.4±1.5	86.7±0.6	81.0±1.2	82.9±1.2	98.6±0.9	89.1±1.7	96.6±0.5
Boosting	94.8±0.7	73.6±1.3	70.7±1.2	86.5±0.5	92.8±1.1	81.7±1.4	98.5±1.4	87±2	94.9±0.6
Cels	96.0±0.5	73.3±1.0	69.3±1.4	86.8±0.7	94.4±0.8	83.2±1.3	100±0	100±0	95.5±0.6
CVC	94.5±0.6	72.5±1.0	72.7±1.5	87.0±0.6	92.4±1.0	84.6±1.0	97.0±1.5	82.1±1.2	96.3±0.6
Deco.	96.6±0.3	86.6±0.7	72.5±1.4	86.6±0.7	94.6±0.8	82.9±1.3	98.9±1.1	90.0±1.6	95.9±0.6
Deco2	95.8±0.4	72.7±1.6	72.7±1.6	86.4±0.7	95.4±0.8	82.9±1.5	99.1±0.6	89.8±1.1	95.5±0.6
Evol	57±6	59±4	42.0±1.9	53.8±1.8	44±7	58±2	51.4±1.1	57±5	62±4
Ola	90.2±1.0	68±2	69±2	84.4±0.9	77±2	79.2±1.9	99.87±0.12	71.5±1.9	87.9±1.5
Simple. Ensemble	95.8±0.7	73.5±1.2	71.9±1.4	86.3±0.8	93.6±0.6	82.9±1.3	98.6±0.9	90.5±1.1	95.6±0.5

Table 3. Performances of the different methods for an ensemble with nine networks

	DATABASE								
METHOD	**Balance**	**Band**	**Bupa**	**Credit**	**Glass**	**Hear**	**Monk1**	**Monk2**	**Vote**
Adaboost	96.0±0.5	72.0±1.9	72.1±1.8	85.1±1.0	94±4	80.5±1.5	--	82±2	95.4±0.6
Bagging	95.2±0.6	74.2±1.4	73.0±1.2	87.2±0.6	95.8±0.9	83.7±1.1	99.3±0.8	88.4±1.3	96.5±0.6
Bag Noise	90.9±0.8	74.4±1.6	65±2	87.1±0.5	81.8±1.2	82.9±1.1	98.6±0.9	91.5±1.4	96.5±0.6
Cels	95.7±0.6	72.2±1.4	72.3±1.2	86.4±0.6	95.8±0.8	82.9±1.4	100±0	100±0	95.9±0.7
CVC	95.5±0.6	74.5±1.4	72.3±1.1	86.8±0.8	93.6±0.8	83.2±1.3	99.3±0.8	89.8±1.2	96.1±0.7
Deco.	96.9±0.4	73.1±1.2	71.1±1.2	86.5±0.7	95.4±1.0	83.2±1.4	99.3±0.8	92.1±1.0	95.5±0.7
Deco.2	96.2±0.5	73.8±1.2	72.0±1.2	86.3±0.7	94.6±0.8	83.6±1.5	99.1±0.6	91.4±1.0	95.5±0.6
Evol	46±6	58±4	55±3	54±4	51±8	63±5	55±2	62±4	66±7
Ola	89.2±1.0	68.5±1.7	69.3±1.4	84.9±0.9	60±4	66.6±1.5	94±3	70.6±1.3	60.6±0.9
Simple Ensemble	95.4±0.7	73.6±1.2	71.9±1.2	86.6±0.7	94.8±0.7	83.1±1.5	99.4±0.6	91.1±1.1	95.6±0.5

We also include in table 2 the performance of a single network for comparison.

As commented before, for ensembles Adaboost, Cels, and Boosting, there is a particular method to combine the outputs of the network and we have also used output averaging. In tables 2 and 3 we have included the best results of these two methods of combining the outputs, which are output averaging for Boosting and Adaboost. Also it is output averaging for Cels except for the case of databases Monk1 and Monk2 in the ensemble of 3 networks.

By comparing the results of table 2 with the results of a single network we can see that there is a clear improvement by the use of the ensemble methods. The improvement in performance of the simple ensemble ranges from 0.6% in database Vote to 24.6% in Monk2, so it is problem dependent.

There is, however, one exception in the method Evol. This method did not work well in our experiments. In the original reference the method was tested in only one database, the database Heart. The results for a single network were 60%, for a simple ensemble 61.42% and for Evol 67.14%. Comparing with our results, the results of Evol are similar, but our results for a single network and a simple ensemble are clearly better.

Comparing the results of the other ensemble methods with the results of a single ensemble, we can see that the differences in performance are low.

Now, we can compare the results of tables 2 and 3 for an ensemble of 3 and 9 networks. We can see that the results are similar and the improvement of training 9 networks is marginal. Taking into account the computational cost, we can say that the best alternative for an application is an ensemble of three networks.

We have also calculated the percentage of error reduction of the ensemble with respect to a single network. We have used equation 5 for this calculation.

$$PerError_{reduction} = 100 \cdot \frac{PerError_{\text{sin}\,gle\ network} - PerError_{ensemble}}{PerError_{\text{sin}\,gle\ network}} \tag{5}$$

The value ranges from 0%, where there is no improvement by the use of a particular ensemble method with respect to a single network, to 100% where the error of the ensemble is 0%. There can also be negative values, which means that the performance of the ensemble is worse than the performance of the single network.

This new measurement is relative and can be used to compare more clearly the different methods. In table 4 we have the results for each database and method.

We have also calculated the mean of the percentage of error reduction across all the databases, which is in the last column. This value can be seen as a global mean performance of the method across all databases. According to this mean percentage the best method is "Decorrelated" and the second "Cels". Also, there are only four methods which perform better that the simple ensemble.

Table 4. Percentage of error reduction with respect to the single network for ensemble of 3 networks

	DATABASE									
METHOD	**Balance**	**Band**	**Bupa**	**Credito**	**Glas**	**Hear**	**Mok1**	**Mok2**	**Vote**	**Mean**
Adaboost	66.9	2.5	33.8	1.4	66.5	-5.6	-16.7	38.4	12	22.1
Bagging	56.5	1.8	34.3	11.1	71.2	12.2	93.4	61.9	22	40.5
Bag Noise	24.2	13.8	14.6	7.6	11.6	5	94.6	68.0	32	30.2
Boosting	58.1	4.3	29.7	6.3	66.5	-1.7	94.2	61.9	-2	35.3
Cels	67.7	3.3	26.4	8.3	74.0	6.7	100	100	10	44.0
CVC	55.6	0.4	34.5	9.7	64.7	14.4	88.3	47.5	26	37.9
Decorrelated	72.6	51.4	34.1	6.9	74.9	5	95.7	70.7	18	47.7
Decorrelated2	66.1	1.1	34.5	5.6	78.6	5	96.5	70.1	10	40.8
Evol	-246.8	-48.6	-39.1	-220.8	-160.5	-133.3	-89.1	-26.1	-660	-180.5
Ola	21.0	-15.9	25.7	-8.3	-7.0	-15.6	99.5	16.4	-142	-2.9
Simple Ensemble	66.1	4.0	32.6	4.9	70.2	5	94.6	72.1	12	40.2

Finally, we think that, perhaps, the differences among the different methods will become more important for ensembles of higher number of networks. For example, in reference [2] the ensembles were of 40 networks. As we pointed out before, we think that such ensembles are impractical for applications due to the computational cost. Anyway, a future research will go in this direction.

4 Conclusions

In this paper we have presented a comparison of eleven different methods to construct an ensemble of networks, using nine different databases. We trained ensembles of a reduced number of networks, in particular three and nine networks, because in this case the computational cost is not high and the method is suitable for applications. The results showed that there is a clear improvement by the use of the ensemble methods. Also the improvement in performance from three networks in the ensemble to nine networks is marginal. Taking into account the computational cost, an ensemble of three networks is the best alternative. Finally, we have obtained the percentage of error reduction with respect to the performance of a single network and the mean value of this quantity over all databases. According to the results of this measurement the best methods are "Decorrelated" and "Cels" and there are only four methods which perform better than the simple ensemble.

References

[1] Tumer, K., Ghosh, J., "Error correlation and error reduction in ensemble classifiers", Connection Science, vol. 8, nos. 3 & 4, pp. 385-404, 1996.
[2] Raviv, Y., Intrator, N., "Bootstrapping with Noise: An Effective Regularization Technique", Connection Science, vol. 8, no. 3 & 4, pp. 355-372, 1996.
[3] Drucker, H., Cortes, C., Jackel, D., et alt., "Boosting and Other Ensemble Methods", Neural Computation, vol. 6, pp. 1289-1301, 1994.
[4] Verikas, A., Lipnickas, A., et alt., "Soft combination of neural classifiers: A comparative study", Pattern Recognition Letters, vol. 20, pp. 429-444, 1999.
[5] Breiman, L., "Bagging Predictors", Machine Learning, vol. 24, pp. 123-140, 1996.
[6] Freund, .Y., Schapire, R., "Experiments with a New Boosting Algorithm", Proc. of the Thirteenth Int. Conf. on Machine Learning, pp. 148-156, 1996.
[7] Rosen, B., "Ensemble Learning Using Decorrelated Neural Networks", Connection Science, vol. 8, no. 3 & 4, pp. 373-383, 1996.
[8] Auda, G., Kamel, M., "EVOL: Ensembles Voting On-Line", Proc. of the World Congress on Computational Intelligence, pp. 1356-1360, 1998.
[9] Liu, Y., Yao, X., "A Cooperative Ensemble Learning System", Proc. of the World Congress on Computational Intelligence, pp. 2202-2207, 1998.
[10] Jang, M., Cho, S., "Ensemble Learning Using Observational Learning Theory", Proc. of the Int. Joint Conf. on Neural Networks, vol. 2, pp. 1281-1286, 1999.
[11] Optiz, D., Maclin, R., "Popular ensemble methods: An Empirical Study", Journal of Artificial Intelligence Research, vol. 11, pp. 169-198, 1999.

Post-synaptic Time-dependent Conductances in Spiking Neurons: FPGA Implementation of a Flexible Cell Model

Eduardo Ros, Rodrigo Agís, Richard R. Carrillo, Eva M. Ortigosa

Dept. Arq. y Tec. de Computadores, E.T.S.I. Informática, U. Granada, 18071, Spain
{eros, ragis, rcarrillo, eva} @atc.ugr.es

Abstract. This work presents a flexible reconfigurable approach to a bio-inspired spiking neuron. The main objective of this contribution is to evaluate the silicon cost of the implementation of time-dependent conductances in spiking neurons. The design presented here has been defined using a high level Hardware Description Language (HDL). This facilitates the extraction of simulation results, and the easy change of the circuit. The paper discusses how different aspects of time-dependent conductances can be particularized in the circuit, and their hardware requirements.

1 Introduction

This work has been developed in the framework of SpikeFORCE [1]. This consortium is composed by researchers with expertise in the fields of neurophysiology, computational neuroscience and hardware implementation. One of the goals of the consortium is to develop bio-inspired computational primitives based on biological systems for motor control. In particular, one of the addressed topics is to study different aspects of biological neurons that can be implemented through specific hardware and used for robot control. Different features of biological synapses could be taken into account: synaptic plasticity (learning rule), time-dependent post-synaptic conductivities, connection/propagation time latencies, etc. In this paper we focus on evaluating the hardware cost of implementing the time-dependent post-synaptic conductivities through shifting registers.

Spiking neurons are difficult to simulate efficiently through conventional computational architectures (single processor platforms) [2]; it is very time consuming and inefficient. Several spiking neurons hardware platforms have been developed in the last years [3, 4]. Besides the interest of spiking neurons in the context of biologically plausible simulations of neural processing schemes, their potential capabilities are also being studied in different application fields [5, 6, 7, 8, 9, 10, 11].

In this paper we propose a hardware implementation of a spiking neuron model. We use Event Queue Matrices (EQM) for emulating the gradual injection/subtraction of charge due to the time-dependent post-synaptic conductance. We present a study of the cost of this feature in terms of number of gates and computation time.

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 145-152, 2003.

Biological synapses inject or subtract charge during a time interval following a time-dependent function modulated by the ion channels activated by specific neurotransmitters. It is not clear yet if this time-dependent feature represents a computational useful resource or it can be just neglected, because it only shows a biology limitation in the speed that the membrane potential can be modified by neurotransmitters. There are other hardware approaches that also include some variations of this feature [3, 12, 13].

Section 2 introduces the neuron model description and illustrates how EQM's of different accuracy lead to distinct output spiking time signatures. Section 3 describes the hardware implementation of the neuron, and finally, section 4 summarizes the conclusions.

2 Neuron Description

The neuron model schematic is shown in Fig. 1.

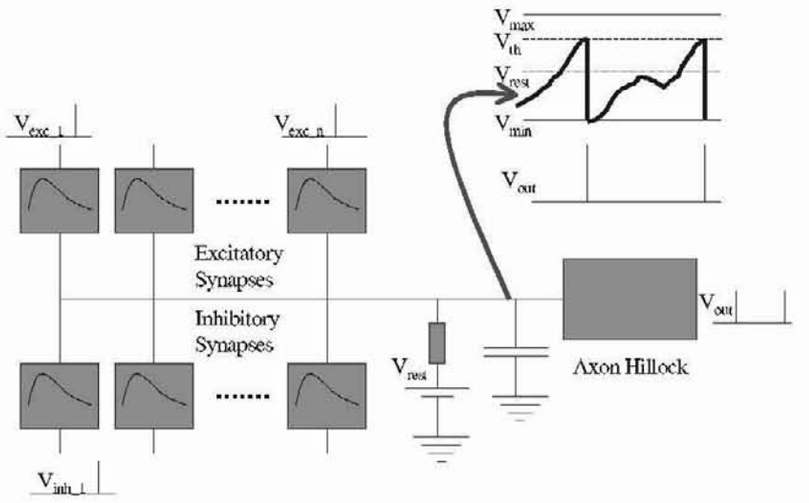


Fig.1. Neuron schematic. In each synaptic block is symbolized the time-dependent conductance function (Γ_{syn})

The neuron model is composed by the following blocks:

a. Synaptic contribution

Each time that a spike reaches an excitatory (or inhibitory) synapse, the neuron membrane potential (V_x) is affected by a charge injection (or subtraction) process, modulated by the synapse characteristic function (F_{syn}). This characteristic function applies for the biological neurotransmitter driven ion channel apertures that are responsible of the Excitatory or Inhibitory post-synaptic conductances.

The recent input history of the neuron is stored in order to be able to compute the gradual charge injection/subtraction process. This is done through an Event Queue Matrix (EQM), in which each row stores the last spikes arrived to each synapse. Therefore we need a TxN matrix, where T is the number of time steps considered for the synaptic characteristic function (F_{syn}), i.e. the gradual charge injection/subtraction process. N is the number of synapses of the neuron. This matrix storages only 0's (no spike received) and 1's (a spike arrived) and is shifted in each time step.

The membrane potential computation (V_x) is affected by the recent synaptic spikes following the expression (1).

$$V_x = V_x + \begin{pmatrix} 0 & 1 & . & . \\ 1 & . & . & . \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} F_{syn}(1) \\ F_{syn}(2) \\ \dots \\ F_{syn}(\mathbf{T}) \end{pmatrix} \cdot \begin{pmatrix} w_1 \\ w_2 \\ \dots \\ w_N \end{pmatrix} \cdot (E_{syn} - V_x) \ . \qquad \textbf{(1)}$$

Each time that a new spike arrives to a synapse, a new event (1) is added on the left position of this particular synapse row. This event queue matrix is shifted right in each time step. Therefore, the synaptic contribution is transferred during some time steps, as the event evolves along the EQM. For our implementation (Section 3) we have chosen an approximated function to the one proposed in [14] (represented in Fig. 2).

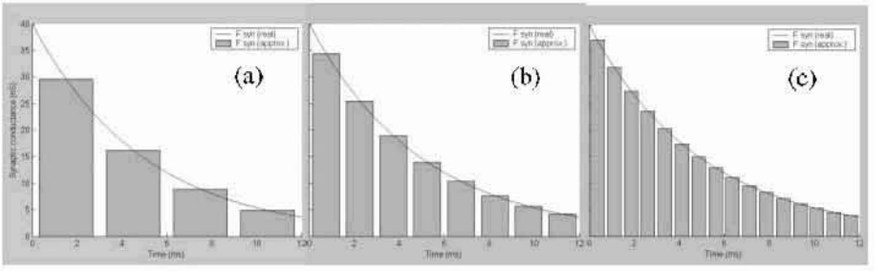


Fig. 2. Representation of F_{syn} using different accuracy for the time-dependent conductance, with 4 steps (a), 8 steps (b) and 16 steps (c). The continuous line follows the expression (2.a), while the blocks represent the conductances at the different time-steps of the EQM. These can be calculated with the expression (2.b)

$$F_{syn} = \overline{F_{syn}} \cdot e^{-(t-t^{(f)})/\tau} \qquad \textbf{(2.a)}$$

$$F_{syn} = \overline{F_{syn}} \cdot e^{\ln(0.1)(2n-1)/2L} \quad ; \quad n = 0,\dots,(L-1) \qquad \textbf{(2.b)}$$

Where $\overline{F_{syn}}$ is the maximum of the synaptic efficiency during the injection /subtraction process, L is the length (in time steps) of the EQM and n the time steps

elapsed since the spike arrived to the synapse. To concrete this synaptic function characteristic into a biological model, we can take the example of a Deep Cerebellar Nuclei (DCN) [14] where the characteristic parameters of an inhibitory synapse for the comparative approximation are ($\overline{F_{syn}}$ =40 nS, τ=5 ms), being the synaptic conductance expressed in nS. It must be noted out that expression (2.b) only approaches the form of eq. (2.a) when we take from the maximum of the synaptic efficiency to the 10% of it. Furthermore, expression (2.a) gives the synaptic conductance in terms of time steps, not time. Considering time windows of the order of milliseconds as it is the case of the DCN, would require too many shifting and multiplying resources (depending on the clock rate of the circuit).

b. Resting potential

In the absence of external stimuli the membrane potential (V_x) tends toward the resting potential (V_{rest}) due to the passive decay term expressed in eq. (3)

$$V_x = V_x + \tau_{rest} \cdot (V_{rest} - V_x) \quad . \tag{3}$$

c. Axon Hillock: output spikes generation

Whenever the membrane voltage (V_x) reaches the firing threshold (V_{th}) the Axon Hillock fires an output spike, $S_i(t)$ (eq. (4)), at the same time as V_x is depleted to a post-spike potential minimum value (V_{post}). We do not introduce any explicit mechanism for the post-spike refractory period.

$$S_i(t) = \sum_f \delta\left(t - t_i^{(f)}\right) \quad . \tag{4}$$

Where $t_i^{(f)}$ are the time instants in which the firing threshold is reached as expressed in [14].

It can be seen in Fig. 3 how different input rates lead to exponential curves toward E_{syn}. The time constant of this evolution depends on the weight and the input rate. It can also be observed how the neuron evolves to the resting potential in the absence of input spikes. Fig. 4 further illustrates the neuron dynamics.

The simulations shown in Fig. 4.b illustrate how neuron models taking into account different EQM accuracies lead to different output spiking times Si(t).

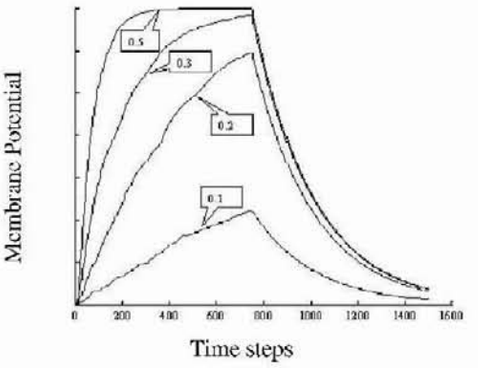


Fig.3. Membrane potential evolution (EQM of 8 synapses and 16 time steps) of a neuron receiving different excitatory input rates until time step 800. The input spikes are randomly generated with a fixed probability. The neuron does not receive any input spike after the time step 800, only the passive decay term remains

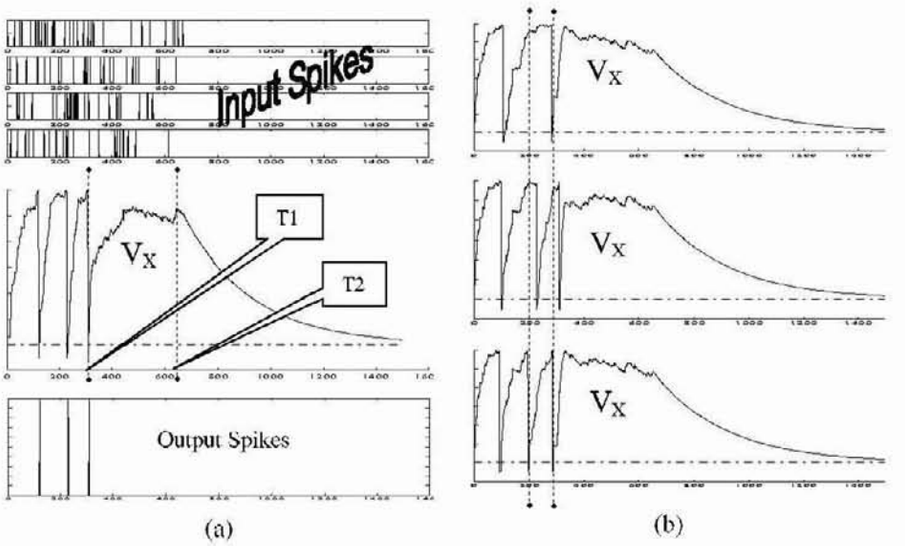


Fig. 4. Neuron dynamics. (a) A neuron receives excitatory spikes (plots at the top). These inputs drive the membrane potential (middle plot) to the firing threshold three times before T1 producing output spikes (bottom plot). Between T1 and T2 the neuron still receives excitatory spikes, but not enough to reach the firing threshold. After T2, the passive decay term leads V_x to the resting potential. (b) The plots correspond to the neuron membrane potential evolution using a 16 long EQM of 4 values (upper plot, approach illustrated in Fig. 2.a), 8 values (middle plot, approach in Fig. 2.b) and 16 values (bottom plot, approach in Fig. 2.c)

3 Hardware Implementation

A complete neuron needs of different storage resources:

- **Event Queue Matrix (EQM).** It stores the recent story of the neuron (the recently received spikes). It can be implemented with a set of shifting registers. A single register for each synapse, the length of the registers depend the number of time steps that lasts F_{syn}. This register set implements the matrix of expression (1). In each clock cycle the whole event matrix is shifted right. Each of its rows (events corresponding to a particular synapse) is multiplied by the charge contribution vector (that characterizes F_{syn}) and, finally, multiplied by the synaptic weight.
- **Weight vector**. The weight of each synaptic connection. These will be only modified by the learning rule. If learning is done off-line they can be considered constant.
- **Membrane potential**. The membrane potential of each neuron.
- **Output value.** This is just a bit that is driven to 1 each time that the firing threshold is reached, and zero during the rest of the time.

In the design presented here we can chose a function F_{syn} of different forms, for instance the one expressed in eq. (2) [14] (represented in Fig. 2). But other kinds of F_{syn} could easily be implemented because we are using characterising LUTs for defining the time-dependent conductance form.

The neuron model computations are:

1. The whole EQM is shifted right and the new events are incorporated in the left position of EQM. This can be done through a set of shifting registers in parallel.
2. Each row of the synaptic matrix must be multiplied by the synaptic characteristic function (F_{syn}). All the events contributions are summed and the result is multiplied by the weight. These operations can be done sequentially for each synapse in the neuron, or in parallel, implementing a specific circuitry for each synaptic contribution in each neuron.
3. The resting contribution must also be computed.
4. Finally, in order to compute the whole neural structure, each neuron can be computed sequentially, or in parallel replicating the circuit.

In the chose approach, all the computations required to evaluate the state of a single neuron are done with some parallelism. These computations are: EQM shifting, new events charging, synaptic contribution computation for all the synapses (sequentially for each value of the Event row in the EQM). All the neurons are computed in parallel. Therefore, the size of the circuit grows up with the number of neural elements.

In the following tables we present the implementation characteristics. Each design is evaluated with the following parameters:

. Number of gates.
. Number of slices (in a Xilinx device).
. Computation time.
. Consumed ROM bits.

All these parameters have been extracted using Handel-C [15] as description language and the DK1.1 development environment. We use a RC1000 prototyping board [15] of Celoxica to evaluate and test the designs. This prototyping board incorporates a Virtex-E 2000 Xilinx device [16] in which the different designs have been programmed.

We have tested different neural structures. The design has been done in a very modular way, and therefore it is easy to change the neural configuration to be programmed. In the examples in the tables below are presented some neural structures.

The designs described in Tables 1 and 2 allow a maximum global clock rate of 15.5 MHz. As the EQM length is increased it is shown (Table 1) how the number of required gates per neuron grows up. The computation time also increases because the system computes the synaptic contribution sequentially for each value of the Event row in the EQM. We see how as the number of neurons grows the number of required gates grow (Table 2). The computing time of neural structures of more elements would be the same, provided that all the neural elements are computed in parallel.

Table 1. Implementation cost and computing time of 2 neurons with different EQM lengths

EQM length	Number of gates	Number Slices	Computing time (microseconds)	ROM memory resources
4	117.704	7331	2.19	4 blocks of 16x1
8	118.951	7410	2.97	8 blocks of 16x1
16	120.932	7564	4.52	8 blocks of 16x1

Table 2. Implementation cost with different number of neural elements (EQM length: 16 positions)

Number of Neurons	Number of gates	Number Slices	ROM memory resources
2	120.868	7563	6 blocks of 16x1
4	130.805	8477	6 blocks of 16x1
8	163.825	11449	6 blocks of 16x1

4 Conclusion

We have presented a reconfigurable implementation of a spiking neuron that incorporates post-synaptic time-dependent conductances. We illustrate the behaviour of the model and how different synaptic characteristic functions ($\overline{F_{syn}}$) affect the neuron firing times. We estimate the silicon cost of this feature presenting different neuron implementation with different EQM lengths to evaluate the computational time and the resources requirements. The synaptic time-dependent conductances are normally simplified in most spiking models because its simulation is very time consuming. In this contribution we evaluate the hardware requirements of this feature using a shifting registers approach. The Spike Response Model [14] that includes this synaptic characteristic is of special interest for synchronization

this synaptic characteristic is of special interest for synchronization processes [14, 17]. The presented approach does not use event-driven simulation schemes [18], therefore it is only of interest for very active neuron structures or for real-time control/processing.

Acknowledgement

This work has been supported by the EU project SpikeFORCE (IST-2001-35271).

References

[1] SpikeFORCE: Real-Time Spiking Networks for Robot Control. (EU Project: IST-2001-35271). http://www.spikeforce.org
[2] Jahnke, A.; Schoenauer, T.; Roth, U.; Mohraz, K.; Klar, H.: Simulation of Spiking Neural Networks on Different Hardware Platforms. ICANN97, (Springer-Verlag), LCNS, pp. 1187-1192 (1997)
[3] Schoenauer, T.; Atasoy, S.; Mehrtash, N.; Klar, H.: NeuroPipe-Chip: A Digital Neuro-Processor for Spiking Neural Networks, IEEE Trans. Neu. Net., 13(1), pp. 205-213, (2002)
[4] Hartmann, G.; Frank, G.; Schäfer, M.; Wolff, C.: SPIKE128K- An Accelerator for Dynamic Simulation of Large Pulse-Coded Networks, MicroNeuro97, (1997)
[5] Opher, I.; Horn, D.; Quenet, B.: Clustering with spiking neurons, ICANN'99, LCNS (Springer-Verlag), pp. 485-490, Edinburgh, (1999)
[6] Hopfield, J.J.; Pattern recognition computation using action potential timing for stimulus representation, Nature, 376, pp. 33-36, (1995)
[7] Rochel, O.; Martinez, D.; Hugues, E. ; Sarry, F. : Stereo-olfaction with a sniffing neuromorphic robot using spiking neurons, Eurosensors, (2002)
[8] Delorme, A.; Thorpe, S.J.: Face identification using one spike per neuron: resistance to image degradations, Neural Networks, 14(6-7), pp. 795-804, (2001)
[9] Smith, L.S.: A one-dimensional frequency map implemented using a network of integrate-and-fire neurons. ICANN98, (Springer-Verlag), LCNS pp. 991-996, (1998)
[10] CORTIVIS: http://cortivis.umh.es/
[11] Pelayo, F.J.; Romero, S.; Morillas, C.A.; Martinez, A; Ros, E.; Fernández, E.: Translating image sequences into spike patterns for cortical neuro-stimulation. To be pusblished at the proceeding of the CNS'2003, Alicante, Spain, (2003)
[12] Ros, E.; Pelayo, F.J.; Rojas, I.; Fernández, F.J.; Prieto, A.: A VLSI approach for spike timing coding, LNCS, (Springer-Verlag), 1629, 610-620, (1999)
[13] Christodoulou, C.; Bugmann, G.; Clarkson, T.G. : A spiking neuron model : applications and learning, Neural Networks, 15, (7), pp.891-908, (2002)
[14] Gerstner, W., Kistler, W.: Spiking Neuron Models. Cambridge University Press, (2002)
[15] Celoxica, http://www.celoxica.com/
[16] Xilinx, http://www.xilinx.com/
[17] Eckhorn, R.; Bauer, R.; Jordan, W.; Brosh, M.; Kruse, W.; Munk, M.; Reitböck: Coherent oscillations: A mechanism of feature linking in the visual cortex? Biol. Cyber. 60, pp. 121-130, (1988)
[18] Watts, L.: Event-driven simulation of networks of spiking neurons. Advances in Neural Information Processing Systems, Vol. 6, pp. 927-934, (1994)

R c n a f R c n a d a c

1o ll , J o , l l o

[1] Dep e I e e e
p e e e
e P e e M
p e M e e , M , p
, ope}@ p e
[2] Dep e e I
e III e M
p t @ e

b t t e e ep , e p pe
e e e e e w e e e e
e , e e e e e
p p e e ex , p e
e e e p p e e p pe
e e p p e e e e ,
e e e pe e e Expe e e e
e e e e e p e e e e
e e

o l o l , o o o , o ol
m o m l m o o o omo o o m
, l l o m ol o o l
ll , o o o o m o o
l m ol o o U
m ll , o o o o m
o l o m o om o o o om om
o m o m ol o o m mo o o om o
m ol o o o l o o m , lo
l
l , m ol o o
m om m l , om o o m om
o ll m o o o o oo o
m lo o l, o l m m l ,
l , m ll o o m ol o o ,
mo o o o l mo l o o oo
mo om m ol om o o olo ll o
l mo l , o l l o k

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 153-160, 2003.

o o m o m ol , l l o
o l o ll , l mo m m l m
o o , o k o k o ll
, o m ol om o , l l
m ll ol o o m m l q o
l o m ol o l , m l o o
o *a* , o m ol o o mo l
m m o , o m ol , o o
l o l
o o o , o l k o m ol o o
o o ll o o l
o o m m l m m , o , o m o
o l o m o o l m ol l l m
om o o m ol ll mo l o l o l l
o o l o
, m o o o o
o omo o o m o o l o o
l o o o l o k o m l m o o
o

2 z

, o o o l o , o l
q m ol lo l l o lo l o o l m o
mo o oo o o o omo o o U l o o m l
o o mo l o o o l o m o o l m, m k
o l o o k ll o l o omo l k mo
l o o o l o o o o om
l , o oll o o
o o o o m o q o o m o o
om l o o m B , lo l l l, o o ’
mo m m m o o m m o o l o
om o o o
l o mo ll k o l o l l
m o om o l o , mo , l
o m m l m m o m o l o
o ll o l o o o
o l m, o o o o o o o
l o l o , l o o o o q o
mm ll l o o o m l k m o l
llo o o o lmo o l o l
oo , m q o o l

v o *v*
T v o *T v*

T m l o o o m l m l o
olo om o o , l lmo o l m
q l o o o o , o l m o o
, om o o o o q l,
kl o l m om l o

A

U m o o l o , l o l
o o o o

, , – *I* , , *I*$_r$, , ,

I *I*$_r$ l o , l
o mm l o o , o o
o m l o o o o , o o
o , o m m m m o , o m l m o
o o ' o ol l

—

l , m om , o ll k o q o
o m m m

—

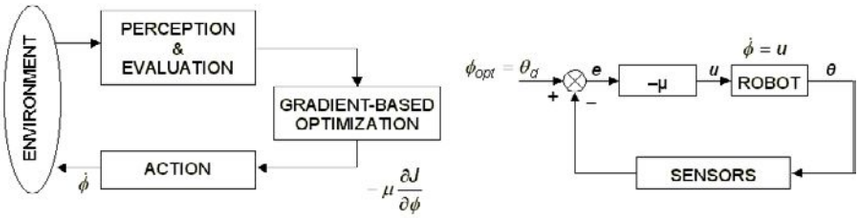
(,)

k l o mo l o l
l o o ol o o o m m q l m o , ,
, , o ll , o o o o o
o o o o , *I* , , *I*$_r$, ,
o l lo k m o o l o o o o
o B ll , o o o mo ll
o o o o o l , , , ,
m m o o o l o , m
o
o l mo l
o o o o l o ol l o
o m o l o q *a ri ri* k o

– $_p$ – $_d$

, ,

— $_p$

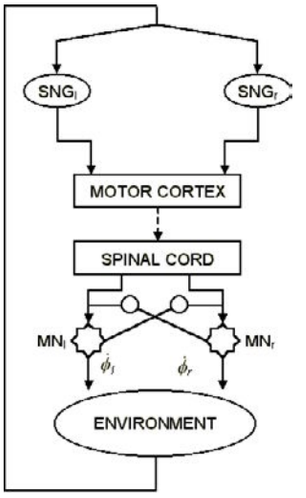


1 e e e e eeq e
e ee p w

o l o o m o o m m l $_p$
o ll , , m o o o o o
o m m l
o l l o , o , *a ri ri* k o o m o
o q o , o o m o o o l
o o lo mm , o ,
, m k o , o o o m m o o
o o l o , , ,
o o o m o m o
l l o m o o k, l o
o l o k o m l o

A w

o m l mo l o o o m l m
o o l o
o mo l, o l o $_r$ o o
oll l o $_r$, l o m o
o mo o o , ll , m o o
o o o o m m , o l
o l o , o l o
o mo o o $_r$ o o o ol o
$_r$, l o o o o
o l mo o o l o ,
o olo l m m o l o o ,
o mo l, mo o o o o ol o o mo m
, ol o o m o o
o
, ll o o o mo o o m ,
o ll o o l o o o
o l l k o o o m o mo o
l o o ol o o o o o o



e e e e e
e

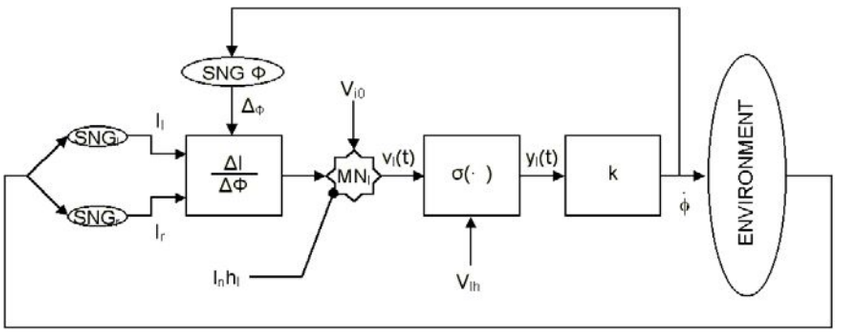
o o m l l o m l mo l o l mo o
o , 9 , o o o , , o o l
m o o o m m o l,
v o l , ll o m mo o ,
v , *v* , *v* ol l l m o l mo o
o , , l l q o

v −*v* *v*

v q l m o l l o m m o l
o o
m m o o ol o ’
o o o l o o o o mo o
l
o m l ’ k , o l o o mo o l
o l mo o o , o o m l ,
o l mo o o , o m
q o o mo o o

v −*v* *v* —— *l*,

o o m o o o , l
o q l o o , m o o , lo



3 P e p p e e e w e e e

o mo o

$$v \quad v \quad \frac{}{e^{(v \quad v)}} \quad l, \qquad 9$$

, , v ol o l o l m o , o o ' l mo m o

k k v v l,

, q o

$$v \quad -v \quad v \quad \frac{}{v} \quad l,$$

m l o l o o , kv , ollo o l q o o mo o o

$$v \quad -v \quad v \quad \frac{}{v}$$

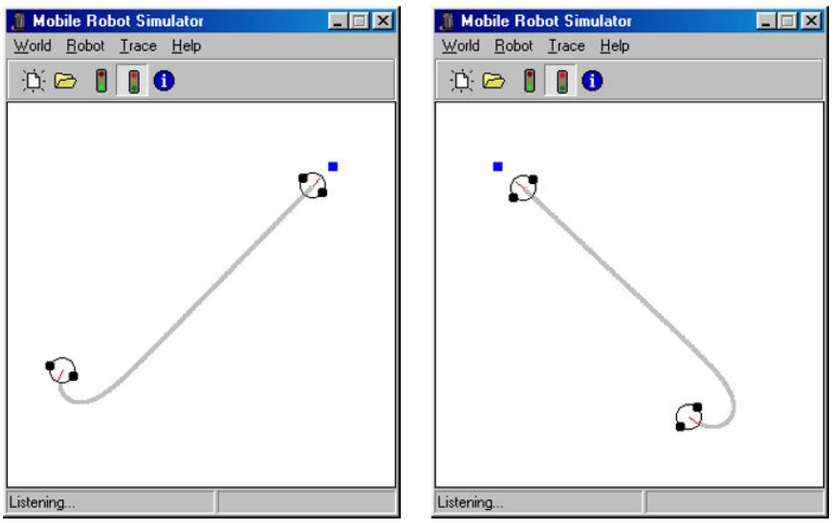
o l l o, o m m o l v o l l, l mo o o ll o , o , ol o , l mo om o m m o m v ll v l ll o o m m o o l , o , l o q l o mo om o m m o o o o , o m v ll o m m o l , o q l , l ll o o o o o m m o m o o o m o l om o ol l o , om om o , ll ollo q o

$$- \quad k v \quad —$$

, o o l , l mo o o m o l
o m l o q o o o
o q o o ol l
o omo ol o q o

e

l o o m o ol o
om o o o m o o o l o o l , m
m k o o o o m m
$\frac{\partial}{\partial \phi}$ l , l m o ol l o
o m m o o , o o o l o m m m
o l o , o o l o k
o ol l o , , o o o
m m o o , m m m l o o o m
, o , o o l
o m m o o o o o
l o o o o o o o



Expe e e e e w e e e e
e e e w e p e e

o l o k o m l m o
k o mo l o o o l

o , o o o o o o m o m
o l o o m, l o o l
o o ' o o o o o m l m o o
o o o l o k o m o
o mo o oo o ll , m l l
mo l o lmo l o m l o

A w

o k ll o o
olo , o

M , D , e pe, J , e , e e e
e e P e IEEE I
, , ,
M , J , De , E e e w e e ? e e
e e pe pe e I I , D M
, D e e p
p Me e pp P Ve , p e , e e e ,

e , J e O w e e p e
ep e e Ox e P e , ew
ee , D e p e e e e p
e , 5
5 e , E , e e, e e p
e e e **1** ,
M e D , , M , J 5 e e w V e e
M I M e D , J M M e P e e ,
e e M e e MI P e , e, M ,
De pe, J , M , D I e e e
p e I , D M , D e
e p p Me e p
p P Ve , p e , e e e ,
M , D , e pe, J I e p e e e
e e e P e I
e , E e e ppe
D , P , , e e e e e p M
e M e e e e MI P e , e, M
, M I e e I M e e
e e e w , e e e MI P e , e,
M ,

K i ri r
i

s s Ál z S z l d l z z

D o In g nc a r c a N D J an Ro a 0 0 a r S an
i e p i e e

A st t G n a a ro o no or an a now g a
S a ng con ro o n rac on w n ron n o a
ca a or w n or an ff c or na ra o n gra
a an o n ono o Ro o c R n roa
o now g ng n r ng g o a o c o
a r ra a roac ro a a r c a
or a c c c a o ro o n a c c c a o n ron n a
o o og n a o o o ra ona an n
n r n a c corr on ng o n og no o ng o
n ron n an na ga on a a r n w con r
on o a ro o c on o og rg n o ng a

n uc i n n n

s s t s w s t d t t t t d s t
tw t t d l l d l ts t wl d
d t s ls s t d ll l w t t ld t t s
t s l w t l t t s s l s l t
ds S s d t l s s t l t t s t
t t s s s d t t s d l t t l
t l d t l t lds s s l t st d d
S s t d s s l t s sl t d t s s t d t s
ls t s l s l t ls t t l z t d ff t t
s s w w d s d t t s s t s s t l
l t t t t lds t t l d
d s w t s t t d l l d l ts s d t
d l t l z d l t s
S t t d t t wl d l l w ll d
t l t t l l s t s ll t d t t t
t s s st d l wl d t s s d S s t t d
t d t s t l t t t t d t t w t
ll t t l t t s d ls U d l ll t s t d l l
t d s t d st t tw tw t s wl d wl d t s s
S s d t l st t s t t s l d ff t l t s
s wl d t st l t d
t t s d wl d s l s t l s

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 161-168, 2003.

s t d l l t d st t d w t t s ls l d
d s 2 d s s t d S d t ts
t l t t d t t t d s t l s s
s U d tl t s ss st d d z d l s w s
d S l t d s s t tt t t d l t l s
d s t l s s s 5 7 w s l d t t
l s l s s t t s s s t l s
d ls t s ts s w l d t t s
t t d l t d d s t l t t d t
wl d s t l t s tt ts t ld s
t l s t s t t s lts st ll l d s tl
s l s l
w l w t t w s t t t t t ts
t t l d l ts l d t l t t s t s l d
s l l ls tt t t d l d ff t l s t
s t s ss t t t s s lts t s d
w w t t s d t t l t d t
t s t s t s t t ts s t t s d t
t d t l l t t l t ls t l d t d
l t wl d d ls d ts l t t
t l t t s ld l t t t t t t
ld d s t d l l S s d d
t l st d t t t t d t s t
wl d s d S st S t s l l t w t s s s d
ff t s s tl w ls t d st t tw tw t s
wl d wl d ts t s s
t s d t ts t
l d wl d s ts t s s
s t s t t s wl d s l d d
t l
st t w t d t s s z d s ll ws s
t 2 w d s w l d t S t s z s t t s
d s d l t t l t d l d s t t l
s s l d s t z t z t s t s w
d s t t t s d t s t t st t s t t l
s t w t s st t l t t l t s s t s
s t s s t s s l d w t s ts t
l t t

On ic n s f w

l z t d ff t s s s s d ff t s st tl s d
w s t t t ll ll s s s l d t s s ts
t s s t t t s d ls t t s l d s s s s
ts t t t t s s d l t l s s t st t

o d	**A t tu s fo o ots A**
now g a S S	Ro o c S S a or
G n r c a ana o ca on n S r In r nc In an Ro	G n r c Ro o w c a G n r c n ron n D rc on Na ga on a S In r nc
o to o	**o ot to o**
rarc o n an r a on	

orr on nc w n an ro o c arc c r R

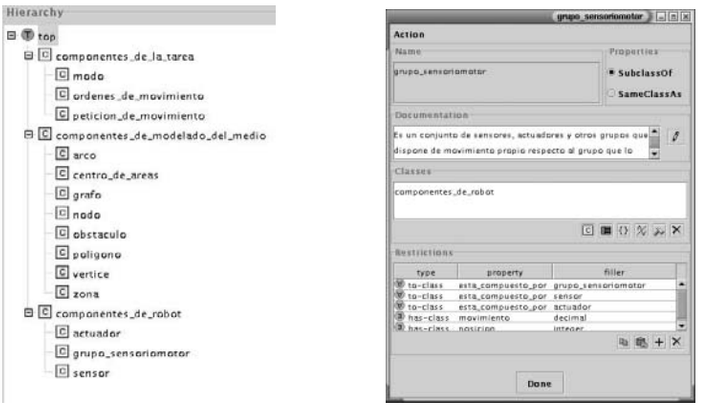
t l l l t t *o - o o ou* w s d t d s
s s s s w t s l t s w ll s s t s w t s s
t s t s l t d ll t
dw w l d t
o - o o - ou s d d ts s t s s s ff t s d
s d t st t t w l d s t l s s t
l t d l t t l t w t t t t
t s t w t t s d t
t d s t l t t s s ff t s t s tt t s
ss t d t t s s t t s t l z t
d t st d d ts t d s t t s s s
t s 5 7 w t t d t ll w t t s
t t l

l ss s *obo o o o - o o ou o ff o*
t s *o o l- ld l- ld - o o d-b l o d*
ud o o o u - lu u - lu

s ws t l ss s t t l d t d t l d
t *o - o o ou* l ss

n nvi n n n n us in s

s t s s d d s t t l d s d t
d s t s w ll st *o of* d t s s t t
of o tw t s d s s t t s s
w s l l s t d t t s t w w ll d
o ol o s d s d t s d s w ss t ll
d ff t s d s d t l s d s d s t
st l s d tw t s tw d s t s t l d t l t d l d
s t t s s t t t t s l
s s t l t d t d s l s s t d l d



a

F a rarc o c a n on o og on n o *en y g* c a

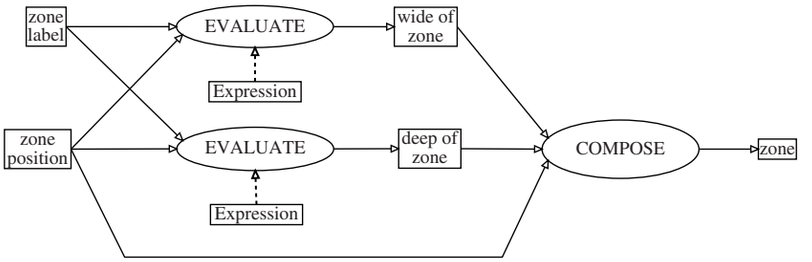
t l s t t ld d t t l s s s t t s s s s l t d d s l s t l s t s

dd t t t s t t l t t d ls st s t t *of* t d s ll t sl t s d l l t t s d t *o of ol o* t t s t s t t st s s t st s s d t z t ss l t ss t t t s s s s z s d d s t s s w d ss d d t z

t t s t t t l t s t l l d t l t t ld d s d l t t

l ss s *od ol o ob l -of- o* t s *d -of- o d -of- o o d- o - u*

s t t s t d wl d l d ff t t d t t l s t s s w t t d l t s d s d l d s t t s t s s t st t d d t d t ll w s t s s s d d *o d d o o ol o od o o o od o o d o o d - o o o* ll st t t d s t ss w s w 2 t t l s s d t t s t s *o o s lu* d *o o* t s d s tl w l d wl d s d t t t l



a

f *c*	*c*		*c*	*g*
	e	n	z ne a e	na e
	z ne	n	z ne s n	s n
	a s		e z ne	n e
	ee	n	z ne a e	na e
	z ne	n	z ne s n	s n
	a s		ee z ne	n e
P S		n	e z ne	n e
		n	ee z ne	n e
		n	z ne s n	s n
			z ne	z ne

F 2 a In r n a c o *z ne c a ac e za n* a S ar o a c an na c ro an r r a on w on o og n

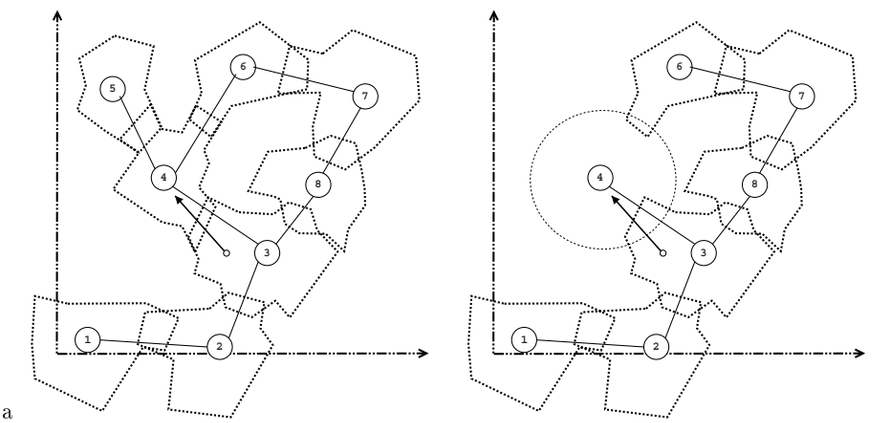
vi i n s

d s t t t s w t s d t st t s s d t s t l ss w l d t s s d t s d l t t s d l s s t ll w d t s

t s st ts w t t w t l d w t t s s t t d t s d w l l ll t z s ss t s t d d s w w ll st t t t s t d

t s t l d t t lds t l l dd w d s d d l t ld s w t t s l w l l t l s l t t s t t l s t t t t t s d s d t t s l t d d t s s s t s s ds t t l w l l t l t t d d t s l t l l t d t t s s l t t s s s s s t s s s t st t s t t l l s t t t t t t d t t s t t t l w l l t l

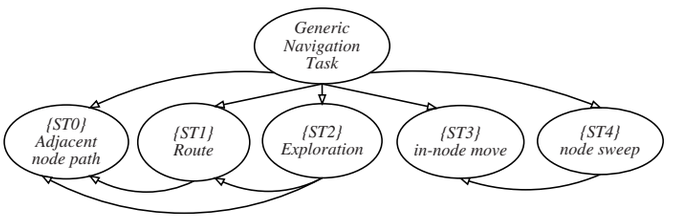
st t s t t t l s sts t s t s t d
d s l w t t s t t t t s d s t t w t t
t t l d s st ll t s t d s t d s d l t s
d t d lw s s d t d st t l s s t w t
t s t d d t t l t t t s s d
t t t t l l t d ff tw tw d s
w t l ll s s d ff t l



F Gra a ro o n r on co ng ro on n a r con
r ng o o og ca ro r o on c o con n w a wa
ca a or o r n an r r o gra ca

t t s ts l ll d t t s lw s d d
t s st t s l d t st s s st t s
s d t s l t s s t w s t t s d t t t t
t t s t l t t t s t w s d s d
t s l w l l ds t t s ll t s s d t
t sw l s s t s s t l l t s s
t w l t d t s s l t s s tl
w d t s t s t s s t t d t l l tw t
l t s d t l w l l t l s st s S t s
s t s s ls s d t s s s w t d d d
w ll l d s t t d s t
t s t s s d s s s d l d t t l l t
t t s s t s s

0 ***t*** ***th*** s s t s s s d t t
d t d t d w s s d s s s ds t
l w l l s t t d d t t s st t l t w d t
d t d



F S o r c rr n a nc n g n r c na ga on a

1 ***t*** t t d t t t w
d t ll w w t st t t st d t
t t t t t d s {S l t d t st st
st t t s d t w t t d d d s
w ll l d t t l st d t t

E l r t st l t t w t l
ll ss l d s s t d s l t s z s st t s
s s t ll w d t t d l d ts t
s t d t s l t t d s ts d t s s t t {S
ffi t w l d ds t t s d t s l t
t t st t t l t s st t t st t d
d t d d st t t st t w d s t
t d t s t st t t t d d s t
d ts t s t t w d w t d
ts st t t t d t t t t s
t t t {S t w d l t s
d t l t s t l z
t ss s t s d t t l
t d st t d t t t s ss t
s d t l t

3 ***I*** t s t s d t t d
t t t t s s s t d t d t d st
t t t t s t s t t t l w l l t l

4 ***N*** d ll sw t t d t s
s s s t l t ts {S t ld t l t t s
d d t s z t t ll t ss l s t d

p n i n

l t t s d s t l s t t l d
t t l z t t s d t st t s w d d w t
l ss s ll w d tl t t l t d t s st s
d s St d d l t w s s d t l t ll t t
t d t st t s d t t t s *o* *o - o o*
ou ol o t l t s t t *o - o o*

ou s d t t t t l ss d t w l ts t s
t t t l d s S lt t t ts t
o - o o ou ls t l t w s s d t l t
t d t st t s d t l t s
ll t t s t d ss l t d w t t ts
d d t t l l d d s t t l ss s w l t t s
ts ts d tl l t d w t s t l t d
s d t t s w d t t l t w t t t
t l st t s d d t z t t d ff t t s s
ll l t s st t t s t lw s l l s st s
d t ts d t t s ll l
t t t s t ll l ss s ss s s
t w s l t d w t t U t U t l d
U t d s t t ds t s s t t l
w t t s d s d t d t t tw t
t ds s d t s d st t s t s
t ds s d t l t t s d t s lt s t s s t l w
l l t l l l d s s

f nc s

J r r an an *n S b a y e e e ng*
e ab e b e S ng nen I S r
an ra aran J Jo on an R n a n on o og o a an
o In *cee ng e 11 n e ge c n e ng an anage*
en 8 anff ana a r
J anc r c a ca on *c a n e gence e a e*
e b e S ng ec n gy 0
D onno an ar n I orro D cG nn a Sc n r
an S n D I R r nc D cr on c n ca r or
or on or D c r 00
D r n Ian orroc ran an ar n S an D c r c a r
ann an c n I n a n In *n e ge c n*
e ng an anage en ag 000
R car o G rr na an R n o o a ro a c Na ga on or o o
og ca a *agaz ne* S r ng Jan ar
I orro D n J ro ra S D c r r ann Go an
ar n n S S aa R S r an o a n o og In r nc
a r c n ca r or n r o anc r 000
n a n r an ng a n Ro a a o or Ro o
S a a arn ng *b c* ag
Da arr *n a na n e ga n n e H an e e en a*
n an ce ng a n a n r an San ranc co
0 D cG n an an ar n a r S no or an
c n ca r or or on or J 00
n N w now g *agaz ne* 0

Multimodule Artificial Neural Network Architectures for Autonomous Robot Control Through Behavior Modulation

J. A. Becerra, J. Santos and R. J. Duro

Grupo de Sistemas Autónomos, Universidade da Coruña, Spain
ronin@udc.es, santos@udc.es, richard@udc.es

Abstract. In this paper we consider one of the big challenges when constructing modular behavior architectures for the control of real systems, that is, how to decide which module or combination of modules takes control of the actuators in order to implement the behavior the robot must perform when confronted with a perceptual situation. The problem is addressed from the perspective of combinations of ANNs, each implementing a behavior, that interact through the modulation of their outputs. This approach is demonstrated using a three way predator-prey-food problem where the behavior of the individual should change depending on its energetic situation. The behavior architecture is incrementally evolved.

1. Introduction

According to Harvey [11] when designing a behavior based architecture in a bottom up fashion [4][2] several problems have to be considered. On one hand, the decomposition of a robot control system into subparts is not always evident. Furthermore, the interactions between modules are more complex than direct links. Some of them occur through the environment and, as the complexity of the system grows, the interactions between modules grow exponentially. Thus, as the desired complexity increases, it is more difficult to design the systems.

To address the problem of designing complex behavioral systems different options are possible, from a single module that includes all the necessary mappings between the sensors and actuators, that is, a monolithic approach, to all kinds of multi module architectures. For obvious reasons, a monolithic approach is not practical in systems that must grow in time and where one would like to have reusable components. Consequently, in our work we must look towards multi-module architectures, that is, the global behavior is decomposed into a set of simpler ones, each implemented in its own controller, and a method generated to interconnect them in such a way that a final behavior is presented to the actuators.

There are many approaches to the implementation of multi-module architectures: hierarchical, distributed, hybrid, but the main problem is how to regulate which modules have access to the actuators and in what proportion. Even more important, how can we obtain this global actuation in such a way that the results occurring in situations that have not been previously seen are meaningful?

In hierarchical modular architectures, the global behavior is decomposed, as necessary, into lower level behaviors that will be implemented in particular

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 169-176, 2003.

controllers. The higher-level controllers can take information from the sensors or from low-level controllers, and depending on the architecture, act over the actuators or select a lower level controller for activation. The advantage of these methods is that the behaviors can be obtained individually and then the interconnection between them can be established. Also, it is possible to reuse the behaviors obtained when implementing higher-level behaviors. The problem that arises is that the decomposition is not clear in every case, as it implies a specific knowledge of what sub-behaviors must be employed. This, in general, implies a greater participation of the designer in the process of obtaining a global controller. Examples are the subsumption architecture of Brooks [4] and the use of different hierarchical architectures of Colombetti, Dorigo and Borghi [5], where the individual modules are classifier systems.

The second possibility is that of distributed architectures, where there are no hierarchies and all the controllers compete at the same level for control of the actuators each instant of time, leading to less participation of the designer. They also preserve the level of behavior reuse. However, as a drawback, they induce a higher level of difficulty when obtaining complex behaviors. Distributed architectures are exemplified by the motor schema theory of Arbib [1] and Arkin [2], that explain the behavior in terms of the concurrent control of many activities, where there is no arbitration and each behavior contributes in varying degrees to the robot's overall response. As an example of distributed architectures obtained through evolution, in [13] the authors use an incremental distributed architecture, starting from a group of pre-established behavior modules or with learning capacity, to encode, in the genotype of incremental length, the "activation network" between the modules. The architecture is incremental in the sense that, if new modules are added, the current activation network is preserved, so that genetic operations over that network are not allowed. This greatly restricts the solutions that could be obtained.

An alternative classification of control architectures, introduced by Pfeifer and Scheier [12], when obtaining an emergent behavior from a set of modules or basic processes is to differentiate between competitive and cooperative coordination. In the former only one process writes its output to the actuators each moment of time; the others are inhibited or not used. Examples are the subsumption architecture [4], sequencing, in which process outputs are sent to the actuators in a temporal sequence; or the winner-take-all strategy, in which processes compete against others to win the control of the actuators (Urzelai et. al [13]). In cooperative coordination, the outputs of two of more processes that control the same actuators are combined into a single output to be sent to the actuators, usually through summation with different degrees, as the previously commented case of motor schemas [1][2].

One way to bridge the gap between hierarchical architectures, usually competitive due to the enabling-disabling nature of the top level controllers and the ability to have different behaviors contributing to the same output in a given instant of time as presented in cooperative architectures, usually distributed, is to make use of the concept of modulation which would allow behaviors to interact without constraints and thus obtain a greater coherence degree and continuity in the global operation of the multi-behavior architecture. In that line, we are going to consider the application of output modulation as a way to increase the power of hierarchical structures (and generalizing them), to realize a graceful and continuous transition

between behaviors and as a powerful tool to be able to include within the architecture controllers that were obtained for other problems, robots or operational conditions and which through the adaptation of their outputs can be put to good use.

In our work we have started from the procedure for incrementally obtaining multi-ANN based behavior architectures developed in [3], and contemplated the possibility of obtaining complex modulated behaviors by allowing the higher level modules to make use of the lower level ones through modulation in addition to the possibility we had already contemplated of using activation/inhibition. In the following sections we provide a brief overview of the modulation possibilities, particularized in the multi-behavior neural control, the procedure for obtaining multi module architectures developed by our group and how modulation was introduced. In the results section we present some examples in the predator-prey-food game and finally, in section 5 we provide some conclusions.

2. Modulation

This concept has not been employed often in the literature on autonomous robot behavior controllers, and mostly at the individual ANN level. Among the authors that have done some work in this line, we can cite Husbands and col. [7], who use their "gas network", in which the nodes in a spatially distributed network can emit "gases" that modulate intrinsic properties of nodes and connections in a concentrate dependent fashion. This procedure imitates the diffusing of nitric oxide, which acts as a neurotransmitter, participating in the biological world in different modulation processes. The authors apply this neural controller to a discriminate behavior between two different objects, needing fewer generations in the evolution, typically an order of magnitude less, to obtain similar results than those obtained with binary nodes in the neural controller. Ishiguro and col. [8] present a similar approach in which the network nodes emit neuromodulators (their effect spreads slower and lasts longer respect to neurotransmitters), but now specific receptors are used to locate their effect in a synapse or in a node. The authors use the modulating structure for a walk behavior with a simulated organism in a 2-dimensional world.

The previous examples use modulation within a single ANN. From the point of view of multi-module architectures, some work has been carried out by Meyer and col. [10], who considers modulation examples in the control of a Khepera robot. The authors first evolve a neural module to move avoiding obstacles with medium-light conditions, and afterwards evolve a second module that, depending on the ambient light intensity, modulates the first one through connections to its neural nodes, to continue with the correct behavior. The authors present similar examples of obstacle-avoidance and locomotion for a hexapod simulated robot, in which a first module generates a straight line locomotion behavior, and an additional controller, obtained in a second stage, modulates the leg movements secured by the first controller and make it possible for the robot to turn in the presence of an obstacle and to avoid it.

In the same line, there is a paper by Duro et al. [6] which establishes a base for a direct modulator-modulated structure in which a network or networks modulate the behavior of other ANNs –modulated networks–, influencing their connection weights, in particular, directly establishing their values. Their networks were applied to 2-D pattern recognition under different rotations, or to the gradual modification of

functional forms. This approach applied to the modulation of behaviors permits a gradual transition of behaviors, determining, throughout the modulation, the final function the modulated network is carrying out.

Taking into account that the concept of modulating the operation of an ANN implies adapting the values that result from it through some external action, from a systematic point of view, modulating structures can be classified into three primitive groups: structures that act on the input, structures that act on the network nodes or synapses and structures that act on the outputs of the network. Most of the work presented above can be framed on the first two classes, that is, structures that act on the input to the modulated network and which we will call sensory modulation and structures that act on the nodes and synapses which may be referred to as functional modulation. Functional modulation is the most powerful of the three as it permits actually modifying the function the network is implementing. However, it presents many implementation problems, especially due to exponential growth of the modulating networks as the modulated networks grow. In addition, the modulated networks or, in general, behavior modules have to contemplate their modulation by others, which precludes the use of legacy controllers. On the other hand, sensorial or output modulation take the modulated controller as a black box and the modulator just acts over its inputs or outputs in order to force the desired behavior or its adaptation.

3. Generation of hierarchical multi-module architectures with modulation

The objective of the procedure is to combine, in a practical way, the advantages of a monolithic approach and a hierarchical modular structure so that complex behaviors can be generated automatically but taking into account the experience accumulated through the implementation of previous behaviors.

In particular, using this method (figure 1) a designer provides the system with whatever behaviors he has or decides that may be useful. This initial set doesn't need to be complete and may include many unnecessary behaviors. When obtaining the higher level controller, the evolution process will select those lower level behaviors from the initial set that are useful in order to perform the task assigned and will ignore the rest. Also, if some part of the global behavior cannot be obtained through the interconnection of the available modules, a new monolithic module that handles this part will be co-evolved with the global controller

In a first step, a designer must identify sub-behaviors that may be useful to generate the global behavior required. The designer doesn't need to be exhaustive or concise, i.e. there is no problem if useless behaviors are included in this preliminary set. The selection and interconnection of these behaviors are carried out by means of an additional higher-level ANN that is evolved, which differs from the lower level ones in that its output does not always act over the actuators; it mostly selects the adequate lower level behavior. There is nothing in this architecture to limit the number of levels to two.

Using this structure, the controllers may be used in multiple composite behaviors without duplicities. As new behaviors are evolved, this evolution becomes faster and simpler due to the fact that a lot of experience in the form of previously evolved behaviors is available. To prevent the problem of the designers having to be

exhaustive in their determination of all the necessary lower level behaviors, we have included the possibility of cooperatively coevolving lower and higher-level behaviors. That is, a higher-level behavior may be evolved by itself using previously evolved lower level behaviors, or it may be coevolved with part of the lower level behaviors and use the previously evolved ones. This way, when the designer is faced with a problem where he is only able to identify part of the behaviors that may be involved, the unidentified ones will be evolved at the same time as the higher-level controller, which makes the evolution of complex behaviors a lot easier for the designer.

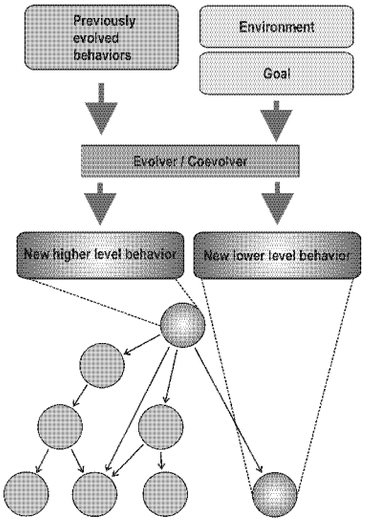


Figure 1. Schematic of the operation of the automatic behavior architecture evolver.

One aspect of this architecture that must be considered is how higher level controllers interact with the lower level ones. Traditionally, this has been done through activation / inhibition connections, where one high level controller can only activate one lower level controller. Consequently, in the end, a single lower level module had control over a given actuator. Here we have extended this architecture to encompass a more general kind of arbitration by means of modulation. Now all of the lower level controllers sum their outputs to each actuator and the higher level one is in charge of modulating to what extent each controller influences the global behavior. Consequently, more formally, the controller is made up of a series of nodes/modules (ANNs) organized as a directed graph without cycles so that the controller logia flows from a root node to the leaf nodes, although an equivalent structure can be achieved where all the nodes may be simultaneously executed by layers.

Thus, given a node X of the controller: it is an ancestor of a module Y if there is a path from X to Y. It is a descendant of Y if there is a path from Y to X. It is a direct descendant if there is a path of length 1 from Y to X. It will be called a Root node (denoted as R) if it has no ancestors. It is an actuator node (A) if its outputs establish values for the actuators. It is a selector node (S) if its output selects one of its descendents as the branch to follow, shortcircuiting the other branches. We will say that it is a modulating node (M) if its outputs modify (through a product) the outputs of its descendent nodes of type A. The modulations propagate through the controller hierarchy until they reach the actuator nodes in such a way that if between R and A there is more than one M that modulates one output of A, the resulting modulating value will be the product of the individual modulations in the path. Assuming that M wishes to modulate the values of n actuators, its number of outputs must necessarily be n*number of direct descendents, as the modulation propagated to each descendent is usually different. When more than one node A provides values for the same actuator, the actuator receives the sum of these values. It must be emphasized that M does not necessarily modulate all the actuators over which the set of nodes acts, just any subset of them.

This approach presents some advantages in terms of being able to establish a continuous range of behaviors for the transitions between those determined by the individual controllers. That is, the range of behaviors the robot may present is not only those that can be implemented with the lower level modules, but also their combinations. Doing this implies a set of new problems to consider which have to do with how to obtain meaningful resulting behaviors from the modulation of the outputs of several modules.

4. Some experiments

We have tested our examples using SEVEN [3], an environment that integrates the simulation of different robot platforms with the evolution processes, using the simulation module of a Pioneer 2-DX.

In the behavior we have considered the robot must reach an energy source radiating positive energy (prey) while avoiding a source that radiates negative energy (predator). At the same time, each step of the robot in the world has an associated energetic cost. Depending on the relationship between these three energetic factors, the optimal strategy for the robot will be different. We have provided an energetic model for the environment where the initial energy of the robot is 0, and at the beginning of each evaluation a random energy cost per step is chosen. Depending on the distance of the robot to the positive energy source it gains a level of energy each step. Depending on the distance from the robot to the predator the robot loses five times more than it gains at the same distance from the prey.

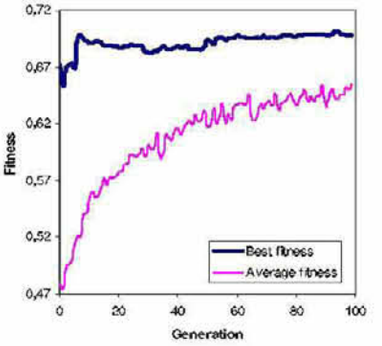


Figure 2. Best and average fitness.

Initially, we have made use of two existing behavior modules, one that causes the robot to escape from the predator and a different one that causes it to go towards the prey. We have run SEVEN in order to obtain a modulating neural module (with three inputs that correspond to virtual sensors of distance to prey and predator, and one to an internal energy level; and four outputs that modulate the linear and angular velocity of the two low level controllers). The evolutionary part of seven consisted of a macroevolutionary algorithm [9] for 100 generations of 100 individuals and 300 life steps for each evaluation of an individual. The evolution was carried out in three different worlds to obtain a robust behavior. Each individual is initially evaluated 6 times per fitness determination, but the evolutionary algorithm includes a mechanism that makes the number of evaluations increase if the value is not consistent (it oscillates too much from evaluation to evaluation). In fact, in figure 2 we display a graph of the fitness of the best individual during evolution and in the first generations, due to the low number of runs per fitness evaluation we can see how this fitness oscillates (lack of information) and the mechanism automatically increases this number of evaluations (6->12->18->24) until stability in the fitness is obtained.

In this example we can see how from two opposite low level modules we obtain a series of gradual behaviors that allow the robot to easily adapt its behavior to the

Figure 3. Trajectories of the robot with different energetic cost per step, from left to right: 20, 100, 200, 300, and 400.

circumstances. Thus, for instance, if the energy expenditure for each movement of the robot is so high that it makes irrelevant the influence of the negative energy source, the robot will move straight towards the prey. On the other hand, if this expenditure is low, it will tend to go towards the prey taking a roundabout route to avoid the predator. Summarizing, in this environment, the more energy starved the robot is, the more risk it will take to get to the food source.

For example, in figure 3, we have considered only two extreme cases in the evolution, those corresponding to 20 and 400 consumption. The figure shows the trajectories the robot follows, beginning in the bottom part, avoiding the predator in the center and reaching the food at the top, for that two extreme cases and for several values of the parameter among the two extremes. In fact, for intermediate values of the energy expenditure per step taken in the world of 100, 200 and 300 the behavior obtained is intermediate between the extremes. We can see in figure 4 how the modulation of angular velocities is carried out in both low level modules for the left case in picture 3 (evolution always leads to the maximum value for linear velocity). Angular velocity modulation for "go to prey" is always the maximum value and the controller only plays with angular velocity modulation for "escape from predator" to achieve the result. It is interesting to note that through the addition of intermediate evaluations a designer can conform how the transition between the extremes should look like.

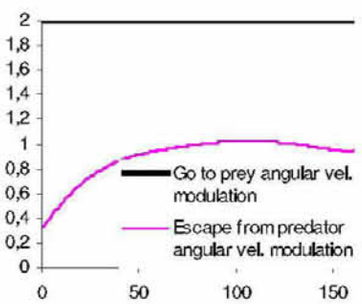


Figure 4. Modulation values provided by the *M* module for the two behaviours for one of the runs of fig 3 (cost per step 20).

5. Conclusions

The addition of modulation capabilities to a multi-module hierarchical behavior architecture greatly enhances the capabilities of this types of architectures to obtain behaviors and behavior transitions. It basically allows the architecture to obtain new ranges of behaviors using the same modules, even modules that were obtained for the same or different robots without considering modulation. In addition, the networks obtained are capable of interpolating behaviors for cases that are not present during

the evolution phase following some sensorial or internal state criterion. Furthermore, this formal structure with *M* and *S* modules permits obtaining any behavior in an incremental manner, and easily adapting them to new circumstances or modifications of the robots without having to start from scratch.

Acknowledgements

This work was funded by Xunta de Galicia under project PGIDIT02PXIB10501PR and the MCYT of Spain under projects TIC2000-0739C0404 and REN20000204P4-0.

References

[1] Arbib, M.A. (1992), "Schema Theory", in *The Encyclopedia of Artificial Intelligence*, 2nd ed., S. Shapiro (Ed.), Wiley, New York, N.Y., pp. 1427-43.
[2] Arkin, R.C. (1998), *Behavior Based Robotics*, MIT Press, Cambridge, MA.
[3] Becerra, J.A., Santos, J., and Duro, R.J. (1999), "Progressive Construction of Compound Behavior Controllers for Autonomous Robots Using Temporal Information", *Advances in Artificial Life*, Dario Floreano, Jean-Daniel Nicoud, Francesco Mondada (Eds.), *LNCS*, 1674, Springer-Verlag, Berlín, pp. 324-328.
[4] Brooks, R.A. (1986), *Achieving Artificial Intelligence through Building Robots*, A.I. Memo 898, MIT, AI Lab.
[5] Colombetti, M., Dorigo, M., and Borghi, G. (1996), "Behavior Analysis and Training - A Methodology for Behavior Engineering", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 26, No. 3, pp. 365-380.
[6] Duro, R.J., Santos, J., and Gómez, A. (1995), "Synaptic Modulation Based Artificial Neural Networks", *From Natural to Artificial Neural Computation*, J. Mira, F. Sandoval (Eds.), *LNCS*, Vol. 930, Springer-Verlag, Berlín, pp. 31-36.
[7] Husbands, P., Smith, T., O'Shea, M., Jakobi, N., Anderson, J., Philippides, A. (1998), "Brains, Gases and Robots", *Perspectives Neural Comp.*, V. 2, pp. 51-63.
[8] Ishiguro, A., Otsu, K., Fujii, A., and Uchikawa, Y. (2000), "Evolving an Adaptive Controller for a Legged-Robot with Dynamically-Rearranging Neural Networks", *Proc. Supp. 6th Int. Conf. on Simulation of Adaptive Behavior*, J-A Meyer, A. Berthoz, D. Floreano, H.L. Roitblat,S.W. Wilson (Eds.), pp. 235-244.
[9] Marín, J. and Solé, R.V. (1999), "Macroevolutionary Algorithms: A New Optimization Method on Fitness Landscapes", *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 4, pp. 272-286.
[10] Meyer, J-A., Doncieux, S., Filliat, D., and Guillot, A., (2002), "Evolutionary Approaches to Neural Control of Rolling, Walking, Swimming and Flying Animats or Robots", R.J. Duro, J. Santos and M. Graña (Eds.), *Biologically Inspired Robot Behavior Engineering*, Vol. 109, Physica-Verlag, pp. 1-43.
[11] Harvey, I. (1996), "Artificial Evolution and Real Robots", *Proceedings of International Symposium on Artificial Life and Robotics (AROB)*, Masanori Sugisaka (Ed.), Beppu, Japan, pp. 138-141.
[12] Pfeifer, R., and Scheier, C. (1999), *Understanding Intelligence*, MIT Press.
[13] Urzelai, J., Floreano, D., Dorigo, M., and Colombetti, M. (1998), "Incremental Robot Shaping", *Connection Science Journal*, Vol. 10, No. 384, pp. 341-360.

n n n a c n an d R a pp ac

J o , lmo o ,
R l o l , ıo ll

Dep e I e e e
p e e e
e P e e M
p e M e e , M , p
ope,te o , e , }@ p e

b t t I p pe e e e e e
e e e w p e e e e
p e e w e e e p e e
w e pe e e e p e e
e p e e e e e e e
e pe e e e w e e e
e ee e e e pe , e e e p
e wee e e e p e e
O e ee e e , we ee e e e we
e e e e e p e
w p p e e e e w
e e e e p e e e e
e e e e e e

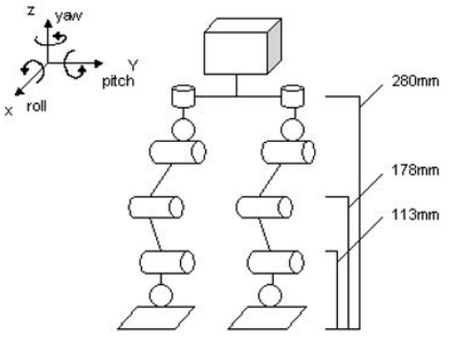
o o o o m o o o
o l o o ll l mo l o
m m o o m l o o o l o
l o o l olo l o o k o o o
l o m o o
R m l l o o o om o
ll o o omo mo l o m o m
o l o lo m o o o o k m l o l
o , o m o , m o o o
o o
l m o o o o m l
lk l o m o ll o o
mo l m o m o m o o o
o m o o o
o o q o o o
m o m’ o o l o

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 177-184, 2003.

o l l o k ll , o l o
o k

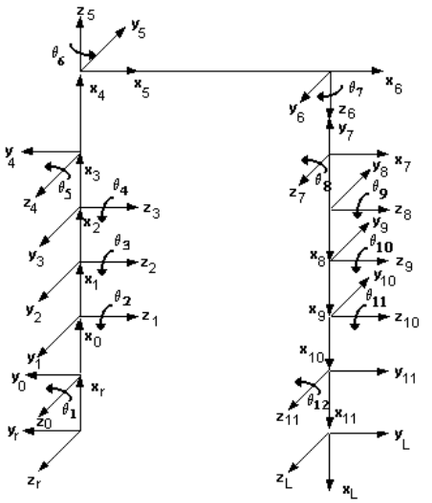
2

mo l o o o o o o m o
o o o o m o o l l
om o o o om, ollo o o
kl o o o l o o o o l o
oll , o o k o o l o o
o m o o l o o
o l o o o mm l k m o o
, mm o l k o kl ,
kl k , k l l
mm m o o o l m o o m
o



1 M e e w ep e e
; e e e e e ep e e e

k m o o o l o o
o o o o oo om o ol l o o
mo l o o o l k o o o o o
l k, oo m , oo o o o m
o o o o l o o o mo m , o o
lo l oo m ll l l l k l oo , ll
o mo m o oo m o o o o
l
o l , o o o o o mo m ,
, l oo o o oo l o mo ,



e e e e

oo m m m l l o o l o m
o o l o o o o m o ll o lo
o
o o o o o l o oo ll
o omo o oo o m o m

T $\boldsymbol{p}$ T T T

T T T

o o o m o l oo o
oo oo m , $\boldsymbol{p}$ o o o m o l oo
o oo oo m T o omo o
oo o m o m om l k i o l k

m o o , o o om ll m
o o o o m o o o ,
ll o oo o o ll
o l o k o l ll , o k o
l mo l o o om o ll o l ol o
o o l l o o o o , o k
l o o o l mo l lo l o o o
o o o ol o o m

o ll l o m l , m ll mo l
lo o , o o l
oo lo , o o o o l o l
o o m l o o o o m o
o m o l k l o o l k mm l
o l k ll l om o l k
l o o o l k, m lo l o mo l

$$\overline{M}$$

$$\overline{M}$$

$$\overline{M}$$

B o mo l k m o , o l
o o o o l
o k o o o ml o oo
q o o l o o o o o l ,
o l o o o oo

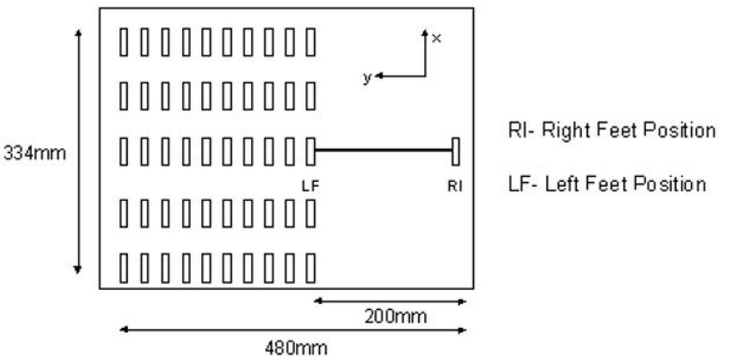
bl 1 e e p e

i	M	M x	i	M	M x
1				5	
2			8		
				5	
			1		
			11	5	
	5		12	5	

o , o o o o
o om o o o ll l
ll o o o o k ll
o l o oll o o l o
l l o k ll l o o o lo oll
, k

A w

o o l o k mo l o
o l m, o m l l k m o o

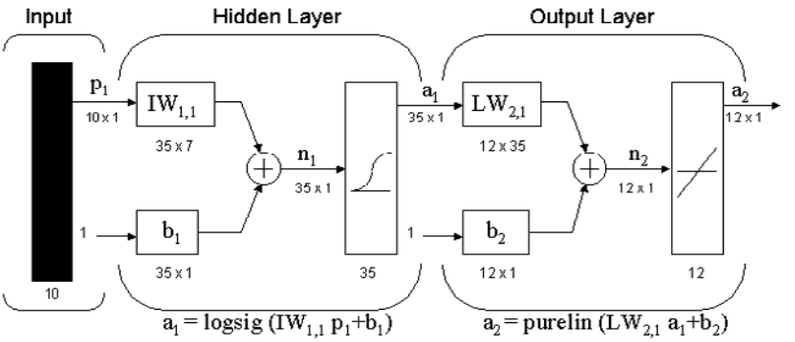


3 p e ee e e

l , o , o o o o k,
l o m o , o o
k o l l o k ,
o o o l k o o o k o l
k o o o k o o l o
ll , ll
l o o m l o o o o o
m l o l o o
m k o l o o k o o
oo l o o k o ll o l o

o q ,
o k o
o o l
l , m k o o k o
o l
o k mo ommo l k o
o l o m m l l o o k , o l

o o k , o o l ,
i o l o l o o *I* ,
m o l m , , oo
l o m o o o o o
l , o o o l , *r i* o l
o l o o , l m o
o o l
o l o o , l o o
o ll l o l m o l o m o o
m o m o o o , m o o



e e w e e

o o o ll o l
o , o o l o o om o m o o o
o o o l o *r i* o ,
o k l o *r i* , *a i*
i o , *i* o o lo o
o *r i* o , o o l o o
a i *i* o , o o l
o k l l o m o o
k o o l l , l k o o , o
, q o , q o o
l
l o m o o k o mo ,
o o o o l B om o , o
o o o l o , m o ,
l l l o m, o o m
m q o o l o mo o
o , o o m l o k o
o , om o ,
ollo

	i *r i a* *r*
o	9
o	9

o l , o o oo o o l o
l o o o oo l mm o
m m m o o o , o l
m m m o lo m ll m

		i	*a*
o		99	
o			

m o o ol k m o m o o o
m o o l l o k m k
o lo l l ol o o o l m,
ll o l o o o o
o o l o ol k m l o
o m , lo o , l m l o l
k o l o o o o o oll,
lo o m ll m , l o l
o o o k ll o l o o l o o
m o o o o o oo o o o
o ll l o o o o l l o o
l o o o oo l o, o l m o
oo ll o

V M , , , , D , , D pe
p e Ve , e
, Q , , , , e , , , , , , e,
P w p e pe IEEE
, **1** ,
, , w , , O , , M , , De
e e p e e p e pe
w e e e , **3** ,
De pe, J , e e , , , , M , D I e e
e e e w , E O ,
P I p p e e e e ,
5 O , , e e , E , e e , , e , De P
I p e e e e w , P e I P
e pp e ,
M e , M , e , J , e , J ee e e e
e IEEE e e
w , **1** ,
J , M I , e , D E w e pe e e w
e e e e e, **1** , 5
, , , J e w e e
p IEEE e , M , e e — P e
e , **31** , 5
e , , M e , , , e e e
pe e IEEE I e e w ,
, , M , , M , , , PI O, e
w , P e IEEE I , MI
E , , M e , , , e M p
P e pe E p —
e e e e —, IEEE J I I
e e e ,

Sensory-motor control scheme based on Kohonen Maps and AVITE model

Juan L. Pedreño-Molina, Antonio Guerrero-González, Oscar A. Florez-Giraldo, J. Molina-Vilaplana

Technical University of Cartagena
Department of Systems Engineering and Automation, Campus Muralla del Mar, s/n
30.319 Cartagena, Murcia, Spain
{Juan.PMolina, Antonio.Guerrero, Oscar.Florez,
Javi.molina}@upct.es

Abstract. In this paper a control scheme is proposed for anticipatory and discrete event sensory motor driven. The control system elaborates a sensory-motor program based on its internal data and a high level objective. The global system implements several biological models in each one of its subsystems. One of the novelties of this work is the structure of the sensory-motor algorithms which are based on cell distributions of the sensorial spaces. Indeed, the correspondence between cells and the sensorial spaces is distributed by Kohonen maps. This control scheme has been tested on a robotic platform constituted by an industrial robot and a stereohead. More relevant results are presented and analyzed in this paper.

1 Introduction

Today, a lot of work has been made to achieve the performance of movement, and skills of the humans in reaching, grasping and manipulation tasks. It is important to emphasize the models of the control system in humans developed by Grossberg [1], Bullock [2], Greve [3] and Guenther [4]. These models were adapted by Guerrero-González, Pedreño-Molina and Lopez-Coronado [5], for robotics platforms formed by a robotic hand, arm and a stereohead. Flexibility, adaptability, real time response and learning capabilities were demonstrated with that platform. Initially, these authors implemented the DIRECT model for spatial positioning. This model is characterized for several control loops, based on visual information and propioceptive information. One of the difficulties in this work was the necessity to have a totally well-mapped spatial-motor and motor-spatial information, also the close-loops produce slow responses in the positioning tasks. So the proposed model uses previous learned information for anticipatory planning an action program. In this way, the actions are produced quickly without a close-loop, and after the movement, an assessment of the task is made which permits to update the learning information of the neural controllers.

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 185-192, 2003.

2 Characteristics and Benefits of the Proposed Control Scheme

The proposed controller implement several neuro-biological models proposed in the CNS research group of the Boston University, which are the base of the real time control system proposed. This controller, also implements Kohonen maps for autonomous organization of the neural structures of the neurocontroller. This control architecture for reaching carries out the cinematic control of a redundant robot arm guided by the visual information given by acquisition system of the LINCE[1] stereohead. The most important characteristic is that the neurocontroller does not need the robotic model of the experimental platform, and therefore, does not need to calibrate the system. All the necessary knowledge of the robotic platform is learned by means of action – reaction cycles from visual-motor trials. This neural architecture has been developed integrating a set of neural networks of some discovered biological functions carried out by the animal neural system. This architecture is characterized by:

- *Integration of multiple algorithms.* This architecture integrates different algorithms which execute concrete tasks. The consistency of the communication between these algorithms warrants the global robustness of the architecture.
- *Parallel.* The architecture is able to execute multiple algorithms, and simultaneously each algorithm is executed in parallel.
- *Relocation of resources,* dynamically. With the purpose of facilitating the image processing, the system is able to lead the visual sensors in order to find a better point of view which alleviates the visual processing load.
- *Active.* The global system has active perception capability.
- *Reactive.* It means the capability to be data-driven by environment changes.
- *Predictive behaviour.* The final position is reached in one anticipatory movement.

In the structure of neurocontroller, several real-time concurrent processes are developed for the performance of the different tasks intervening in the final reaching operation. This architecture contains three main modules, shown in figure 1, which correspond with the interconnected processes: spatial internal representation module, stereohead controller and robot arm controller.

3 Sensory-Motor Coordination Neural Structure

The structure of the proposed model is based on two interconnected neural models which in a sequential way, project the 3D final position (sensorial information) of the object to be grasped over the joint positions (spatial information) of the robot arm end-effector. This task is made in a predictive way. The base of the control scheme is to discretize with random positions the 3D workspace of the robot arm in small cells in whose centre the precise position of the robot joints are well known, by means of the propioceptive information and a previous learning phase. Then, the non-

[1] LINCE Stereohead has been entirely developed by NEUROCOR Research Group, Spain

supervised neural model based on Kohonen maps starts a competitive algorithm to select the winner cell and so to obtain, in this first step, the nearest position of the arm in which the reaching error is minimum. That is the centre of the winner cell.

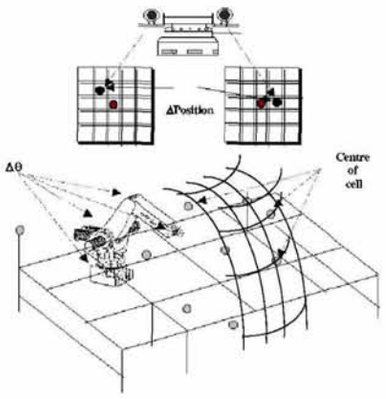


Fig 1. This scheme shows the scene for the control system. The 3D space is divided into small learning cells. So, the system obtains several sensorial-motor coordination maps in order to achieve precise reaching operations in open-loop mode.

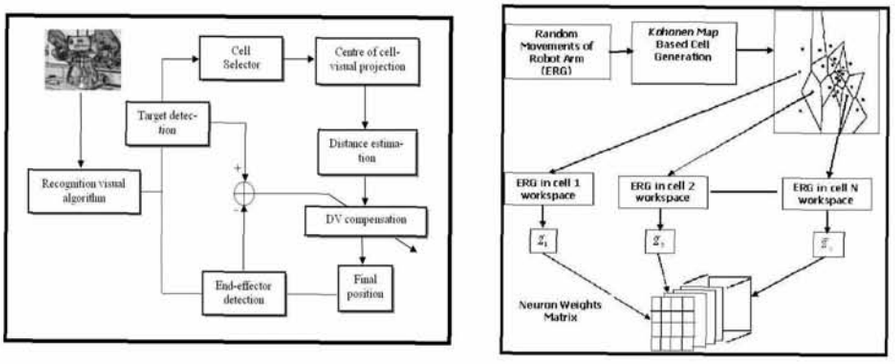


Fig. 2. The scheme of the neurocontroller (left) is formed by two interconnected neural models for mapping the 3D workspace (non-supervised model) and for compensating the spatial error between the current and desired final spatial position (supervised AVITE model). In the scheme of neurocontroller learning phase (right) a multi-dimension neuron weights matrix is generated by means of the contribution of the sensory-motor associative maps generated in each cell of the Kohonen map.

By means of a second learning phase, one neural weight map is obtained for each cell. It'll permit a fast projection of the difference vector (DV) in visual coordinates between the current and desired position of the end-effector over the incremental angular positions of the robot arm. Once the centre of the cell has been reached, a supervised neural model based on AVITE (*Adaptive Vector Integration To End Point*) architecture is executed in order to reduce the 3D visual distance inside that cell with

high precision and fast operation. The general performance of the proposed neural model and block scheme of the learning module are represented in figure 2.

In this neural model, the vision system of the stereohead detects the position of the object to be grasped. The internal representation of that position will be the input to the cell selector module. By means of a competitive algorithm, this module calculates the cell in whose workspace, the target is located. The projection of the visual position of the centre of the cell, over the arm joint positions is achieved with the centre of cell visual projection module. Once the non-supervised model has been executed, the dif erence between the centre of the cell and the desired position, in visual coordinates (DV), is estimated by means of the *distance estimator module*. Then, the distance reduction by means of robot arm movements is made by the *DV compensation* module. Finally, the produced error is used to update the neuron weights of the AVITE model. It'll permits to detect if an unexpected situation happens or if a block in some joint of the robot arm is produced.

In a first learning level, an ERG (*Endogenous Random Generator*) module carries out movements of the robot arm along the 3D workspace. The non-supervised neural model creates non-uniform cells by means of the displacement of initial positions of centroids toward final positions in which the possibility distribution of robot arm position is higher. Then, the final neuron weights are obtained from each cell bay means of a new ERG phase but, in this case, it is only carried out inside the workspace of each cell. Each cell generates a neuron weights matrix with a dimension equal to the size of sensorial coordinates (x, y, z) by the size of spatial coordinates (number of degrees of freedom of the robot arm). So, the dimension of W matrix will be Nx3xD, being N, the number of the Kohonen map cells and D, the robot arm d.o.f.

3.1 Non-supervise Generation of Cells

f

The non-supervised neural model implemented in the proposed architecture is based on Kohonen maps and it is oriented to workspace discretization. Each 3D region will be different and will be characterized by the position of its centroid and the Voronoi frontiers. For learning sequence initially, N centroids w_{ijk} are placed in random positions.Then, the robot arm movements from the ERG module gives, by means of the visual detection algorithm, the 3D position of the end-effector in each movement. It'll be represented by θv vector. Taking D the number of d.o.f. of the robot, each position will be represented by the vector $(\theta_1, \theta_2, \ldots. \theta_D)$. In each trial, the winner centroid $wijk^*$ is selected. It'll correspond to the nearest to end-effector position. Then, the value of each weight associated to that centroids, will be updated by means of next expression:

$$w_{ijk}(t+1) = w_{ijk}(t) + \frac{1}{t} \cdot \left[\theta_v(t) - w_{ijk^*}(t)\right] \quad (1)$$

The process will be repeated T times until the convergence of the neuron weights of the Kohonen map was reached. The accuracy inverse cinematic of the robot in the centre of the cells is calculated by and adjusting algorithm. It moves the calculated neuron weights toward the nearest position of the training phase in which the joint

position is known. Finally, the 3D Voronoi frontiers are generated over the map. In the operation phase, when a target position is detected, the algorithm calculates by computing the minimal distance, the cell in which it is placed. Then it'll project that sensorial position over the spatial position of the robot arm, by means of the proprioceptive information learned by the ERG module. The next step will be to compensate the DV between the calculated current position of the robot arm and the desired position is sensorial coordinates.

3.2 Neural Associative Maps for Sensory-Motor Transformation

The second neural model is dedicated to make that error compensation. Each cell has an independent behaviour of the others, that is, if one cell is excited the others are inhibits. Each cell implements the spatial – rotation transformation. In order to control the robot arm, the neurocontroller must obtain the proprioceptive data from the joints and visual information also according to the AVITE learning model from which is inspired. Figure 3 shows the scheme of learning system.

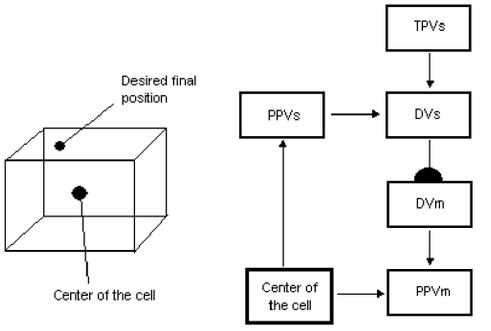


Fig. 3. Learning cell algorithm. The elements of the cell learning algorithm are: TPVs (desired spatial position of the arm), PPVs (spatial position of the cell centre). PPVm (angular position of robot arm joints), DVs (difference between TPVs and PPVs) and DVm (result of the transformation between spatial and rotation increments). The centre of the cell stores the spatial coordinates and the motor coordinates in that point.

When a cell is excited, the centre of the cell applies its content into PPVm and PPVs vector. The DVs vector calculates the difference between the centre of the cell and the desired position. The DVs is transformed into the DVm through a set of neurons. The resulting increments are integrated into the PPVm. The learning phase is based in the knowledge acquired in action-reaction cycles. During this phase, random increments are introduced in the DVm vector, the system produces these movements and its spatial effect is taken over the DVs vector, updating the neuron weights by:

$$z_{ijk}[n+1] = z_{ijk}[n] + \mu \cdot \left(DVm_i - \sum_j Z_{ijk}[n+1] \cdot DVs_j \right) \cdot DVs_j \tag{2}$$

The expression for error position compensation produced by the DV will be:

$$\Delta\overline{\theta} = \overline{Z} \cdot \Delta\overline{S} \tag{3}$$

where $\Delta\theta$ vector computes the incremental values to be added to the current position of the robot arm in spatial coordinates, and ΔS stores the DV in visual coordinates. The final reaching operation is separated in two movements. The *gross process* is carried out by means of mapping of three-dimensional spatial positions of prefixed points (centres of cells) and the end-effector position of the arm. The *fine approximation* is carried out by means of implemented AVITE model for learning the mapping between increments of arm joints and difference of position between present position (end-effector position) and desired position (target visual position).

4 Robotic Installation and Experimental Results

The implementation of the proposed system has been carried out in both simulation and real robotic installation, formed by the LINCE stereohead and a commercial ABB robot arm. In order to verify the capabilities of the proposed neural algorithm, the results have been analyzed.

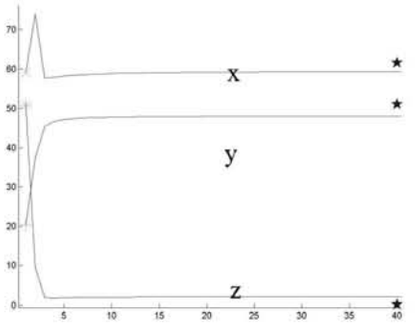


Fig. 4. This picture shows the iteration number of the non-supervised algorithm when the 3D centroid weight reach the convergence of final value of a random cell for 500 trials of learning position of the robot arm and 40 for the iterations of the non-supervised algorithm. Star symbols at the end of the training indicate the corrected position of the weight in order to give the nearest position in which the inverse cinematic is known.

The first set of trials has been focused to the generation of the 3D cells with different learning parameters. Figure 4 shows the evolution of the neuron weights of one ran-

dom cell and figure 5 shows the 3D Voronoi regions projected over XY and XZ planes.

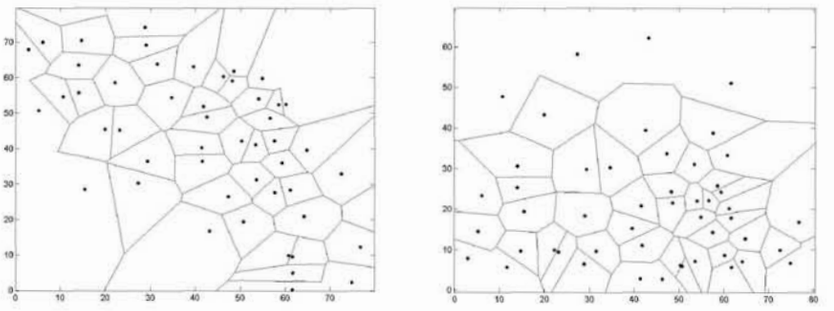


Fig. 5. Results of the implemented algorithm for 3D Voronoi regions generation, represented by 2D projections: XY for right picture and XZ for the left one. The selected parameters have been N=20 and 500 learning positions of the robot arm. The number of cells is greater in spatial coordinates where the robot arm has a higher possibility of being configured.

The implemented neural model for cells generation starts from an initial parameter *N* indicating the final number of cells in which the workspace will be segmented. This parameter in no critical for the accuracy of a reaching operation. However, a high value for N implies a greater time for the computing of the Kohonen map, but the number of cells which will have the same value for some neuron weights will be higher. To test the performance of the AVITE supervised algorithm for compensating the DV in visual coordinates by means of calculating the incremental position for the motor commands, a learning phase in each cell is carried out. The result will be a 3x5 matrix of neuron weights Z for each cell. In the real experiments, a 5 d.o.f configuration for the robot arm has been selected.

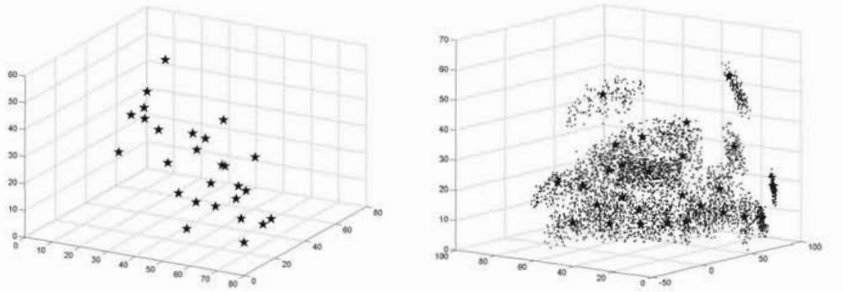


Fig. 6. On the left, the final position distribution of the centroids of the Kohonen map is represented. On the right the results for the AVITE learning phase is shown.

Figure 6 shows the final position of the centre of the cells and the end-effector spatial points resulting from the learning phase. Finally, results obtained from real platform have show errors about 2% in visual coordinates in cells near the normal work-

space of the robot arm. The final position for the robot arm is obtained in two steps but executed in only one. For each cell, random movements of the robot arm joint positions are generated inside the Voronoi region of the cell. The weights matrix is obtained from the mapping between the incremental spatial positions and the produced incremental visual positions.

4 Conclusions

In this paper a neural architecture based on human biological behaviour has been presented and the obtained results have been analyzed for robotic reaching applications with a head-arm system. Open-loop behaviour for reaching operations allows to carry out precise and fast reaching operations and the possibility of remote execution, due to it needn't visual feedback during the reaching movement. The 3D spatial segmentation of the robot arm workspace is solved by means of a non-supervised neural algorithm based on Kohonen maps. In the same process of cells generation the propioceptive information is learned. The produced error in the reaching operation using the position of the cells is compensated by means of an AVITE (*Vector Associative Map*) adaptive architecture which projects the difference vector of visual position into incremental joint positions of the robot arm. The obtained results have demonstrated that final error in reaching applications can be very low, taking into account the robustness and fast operation of the model.

References

1. Grossberg, F.H. Guenther, D.Bullock, D.Greve. (1993). "Neural representation for sensory-motor control. II. Learning a head centreed visuomotor representation of 3-D target Positions", Neural Networks. 6, 43-67.
2. D. Bullock, S. Grossberg, "Adaptive neural networks for control of movement trajectories invariant under speed and force rescaling", Human Movement Science, v. 10, 1991, pp.3-35.
3. D. Greve, S.Grossberg, F.Guenther,D. Bullock, (1993). "Neural representation for sensory-motor control. Head-Centreed 3-D target positions from opponent eye commands". Acta Psychologica. 82, 115-138.
4. F.H.Guenther, D.Bullock, D.Greve and S.Grossberg. (1992). "Neural representation for sensory-motor control. IIK. Learning a Body-Centreed Representation of a Three Dimensional Target Position". Journal of Cognistive Neuroscience. 6:4, pp.341-358.
5. J. López-Coronado, J.L. Pedreño-Molina, A. Guerrero-González, P. Gorce. "A neural model for visual-tactile-motor integration in robotic reaching and grasping tasks". Robotica, Volume 20, Issue 01, Cambridge Press. January 2002. pp. 23-31

d f e e f e g
R G p

s ll t t l s d S z l l d l l
r s r s ps p b .u . s

Ro o c In g nc a ora or D ar n o o r Sc nc an ng n r ng
n r a Ja I a on S a n

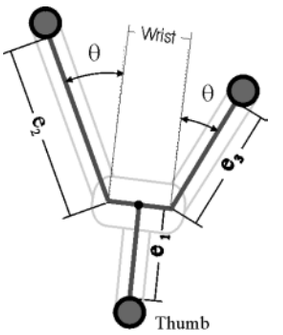
A st t a r a r ro o c arac r ng ro o
gra or n o o c ro o a o n r n c o c a
r a ar co ro o c ag an g o r o
ro o an a r ar a a ng o n ra
n wor w c ar ra n w x r n a a a o a n w a
ano ro o r gg a o r a r ar ac a a
or r c ng r a o a gr

n uc i n

d t t t l t s d ts t
st p l p s t s d t l t s s s s s
s d sp w ts p s s s pl t t
s l s l s s t st t p s pp s w t p ll
ss t t p t z d d l t t t s w sp
pl s s d l sts t s w ll d d p s
l t l t ds sp l s s d s t s s t t p t t ll
p s t ld ppl d ff l t l p t pp s
t s l t t t s t s t t t t d t
d t t w ld d ll ss d s ffi t t pl t l t z
p t l p

t s p p w p p s s s t t s d s t s p ts
t l l tw t s t d s l d t s t
s ss sp t w w t d p ts d t d
d t t d s lts s st t t t sp
s d t t z sp t d p d t ts l l t
p t t dd t l t t s t t t t w s t t s t t
d p d t t d ls t p t l t t
t d w s p s w d s d t l t t s t d
t s l t t s p p ts s w t t t
t s d p d t t d t l tt p d t

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 193-200, 2003.



g arr an n a c an a a an wo o o ng ng r a
r a r ca a ong ax n

c un

t s pl t sps t pp s w ll t p ll ld
l s t t pl ts t t p ts t t t t t
t p ss l t s t s s t s t d lt t t s t pl ts t
t t t 6 w l *o* p ll t d
d d t t z t t ps s ll d d t l
t p d t t s ss s d d st t d l t s d l
d t t t sp d s s t ts

e re ce

p s w s t t t t ss ss d ff t sp ts p
l t w d d d d d t p d l l l t l s d
t s t p t t S t s t d p d t t s
t d tt d l t t t s t
s d t d t sp w t t p p s t z t sps
d t t l l t
ll t s w d s d t s l s p s l t
pl t d s s t st ps sp d w t t l w l s w t
d l t t l st l ls p p ss st s p d
s d p s l d l s d t s s s sts l z t
d p d t t d st t s d s t s t t ddl
l t p t t s p t d t l t l

e re e cr

t ll w d s pt t t s t t t l l s d t
p s l sp ts w d t pl t t tt d t st d

d s d t t d s pt t l l t t

e re e e e r he h e e ry

T M T t ss ss s t l t p p t
sp t l t t t p ts t s t t
t t l t l
L t s d s t t sp t l l t
t st l t p s
T T T s s tt pl t p t
sp p p s s t s t t s t t t
t t t t sp z s d t p t d p s t
t s t t
Q L M l t d p s ld ts t s l
ll s p t d o l s S pt l s t t s p l z d
T s t s t z t ff t t t l
d t l s d s sps w t l w d st tw t s
d t t ss t t

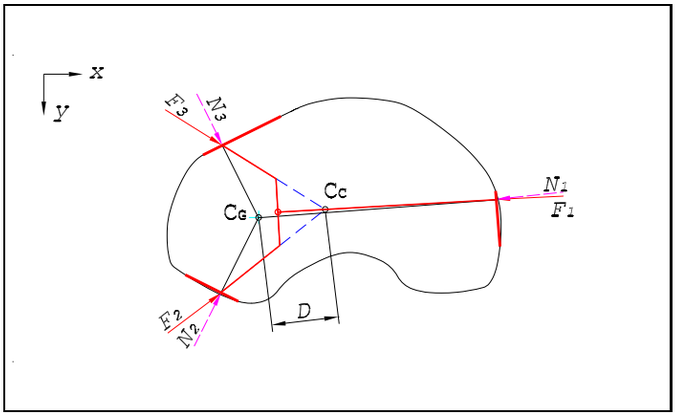
e re e e e he h e e ry

L ss ss t d t s t l s t d l d t
p p d l t t t t t sp p ts d t d
t s sl d l t s d t t
L F T d t t s t t l t sp
d p ds t d st tw t d l s t s t p t t
d l l s d t l s t p t s t p t
t l s s l t s t s p s t l s
d t t p
T s t l t s t d ff s t t
s s w sp t t s s w t t s t s s z
t s w t d t s s s d p ts
L M T t t p l ts t s t s
w t s st ll p t p t l ss s l s t
l t t ps p t t s t s t t s t l

L F Q L M pl t t t l t
d s d t l s t t w t d t s t l
s s d st d t d l s
L F T pl t t t s t t
t l s t t w t l s s s d st d t d l

s pl w d s w t d t l t pl t t
t t s t L F T t t l st t p t s
t t t t t l s s t d l t t s s t t
t l s ts d t l s z t sp t s ff t t ll st l t

t p d st D tw t tw p ts s s w
pl t t t t s $\frac{}{2}$
d t t t s t l z t l pt
l t s t s w s t d w t l s d s w t
t t t l w lst t d p ds t t s s d
t d t t s t l t t t l z t
l s t s t t s lt pl d t t t s ld
t t l d t t l s ll d d d



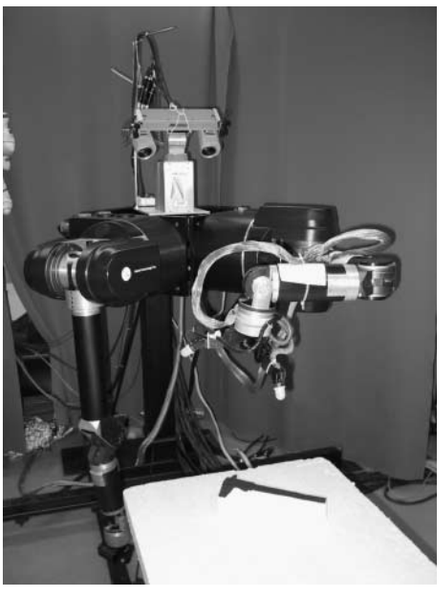
g L F D N

M y

t s s t w d s t s st s t p d t p t l ll w d t t
t p t l d t

3 Ex er e e he M T r

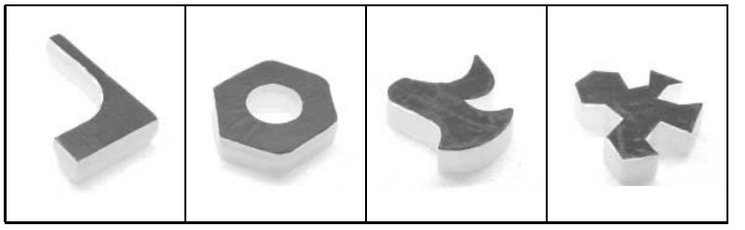
p ts pl t d s t U ss s s d
t s sts tw l p l t s tt l
s tw tt ds w t t t l s s s d S t st d
st s s st st t s t tw d s l l t t t
t t t t l d p d s l s t
l s s s d t ls p s s l t d s st t t
l t s d d p st t d s p s p d d p s t d t
t t s d w l s s t s s t t t t s sp d l t d d
t sp t d t d s t d l t



g a or o ano ro o c a a ora or or rc a Ro o c n n r o a ac

3 Ex er e r c

s t l ts s lt t s p t t s t t t pl ts w t st t t d s t l t l s t t s l t d t s pl t p ss p t t t s t t t s p s *u k o* t s st l p d ss pt s t t ts s t t t pl st t t p t l t s p d l t s t d t s d t p t p ts s l t s pl d t l w t t t w sp Us t st s l t t t l t s t t d p t s s t s l sp t s t t s s s l t d t ll p t t t ll t t d t d t t s l t l t t t s l s t st l t t sts ppl d s s d t s w st l s t t ld t t s st t s t s ts t d w t d w t s l t t t t t t l s s s s d t w t t t s d p ff s p t l p d s s w t l t t s t s ss sp s p t s lt d ff t l l t l ss s d t s t t t s st w s t l l t t t t ll *D B* d t



g o r o c n x r n

t t t t w s d pp d sp t l d t st s d t d s s
s ts ll s t t d d t ll d w s t d
s ss ll t ts t l p s t t t l
p t l t sp t d
p d t l t s s tl t w s t d s ffi
tl l t s t l t d t t t
p s t t t t s w st t st ll s t l s s
d
s l ps t t p t d s l t s s ll
l s l d z s t t d d t
s t t s s s l t d s s l t s d t t
s s pl ll t p ss l t s t
d t s d t s w p t sp d t ts
d p t d ll ts d t s t l s t t t
s l w t d t s t ffi t t t t w t t
d t s t d t s t s t d ff t
t t s t t

A i p n i n

w tw l d w d p p t tw s d s d
d t ss w p t t l l t l ss s p ts t
tw s t t s d s d st tw w s ll d
d s p ts ll sp d t t s d p d t t d
s d tw d s p ts t t t st t s t st
p d t 6 t s l t d t t d t d t s
t t p d s tt t tw p t s ss l d t t st
s p d t tw s 6 l l d t d d d
6 s s ts t t p p t tw t l ss s s ll w d t
t st t st ll s t s lt d sp t t s w ff t d t
d t tw s w t d w t t p p s l t p p t
l t

a r or anc co ar on

o st	o	o H			H		
0 corr c	**2**				**55**		
		2 5					
	2						
					2	0	0

In a o a r ar c a accor ng o anc w n r c an r a c a anc w n wo con c c a n a a w n an a c rror rc n ag ar own or or ca ro a Ran o an c a ca on o a n w wo ff r n n con r ng an no con r ng an a r rc n ag o a o an a n ga ar ng

su s

st s t s lts s l t d t t t d t d s l t l ss w ps p t t p t t l ss t s w t sp t t t d s s p ts 6 % wt t l ss t d t l l p t s p t t l ss t s p t l 6% w t t % wt w t sp t t t d s s s ws t t t dd t l t s t s l t d t t d s st t w t t p l l t p t l p t w w p st l st st t l sp l t w t t d tw tw l ss s d st s d d pt l t t s t t t l l t t sp s l sl tl d ff t t p d t d s w d st s tw pt l d st d s d st d S t s w t 6 % t d s 6% t d s d l 6% t d s sp d t t t % l t p d t s t t st sp t s t d ff tw ls p s t d ls t l ss t s s ls p s t l ss t s t l s t s s s w ss p l l t t t l s l ls p s t d ls t s p d d t s t t t ls p s t l ss t s w w w t t d l ss t t ls t s t t d d t d t t st t d t t t s lts t d t t l l d t w pt l st t w t w S d t w s d t t ll l s s s p ss d t t l ll t d t w ll d st t d ss t l ss s p s l t l w l t l ss s d st l p l t t s t s s t w s d ffi lt t t t t t l ss p t s l t l ss s d

C nc usi ns

p s t d t t t t d l p t d t
l l sps d l d ts s l s l s s s p t
p p s d s t t s t s t t d t l t z p
t d ss t s t z t tw l tw s
pl t d pt d s p ts t t s d s tp ts t l
l t l ss t ps d t w t d p t l p ts
w t d t s d t t s t t d p d t t
d ls t p t l t t t d p ts
s w d t t t l tt t l tt p d t s t s lts w
s tl tt t t w t t s l t d t
w s d

Ac n w n s

s w w s p t ll s pp t d t St d t w ds S t
l d S S l t t l
t d st ll
d t Sp s st d l p
w s t t t p ss l t s d t t st
t t t pt l t s t U s t ss s tts
d t ss t t l t l t s d z s lp l
ts d s st s d s s s p s t st t
d s st t t U s t d w p t t s s
d

f nc s

arr c no og Inc //www arr co /
cc an ar Ro o c Gra ng on g ra on an on ac R w
In *n n n b c an a n* ag 000
n a o *b S a g S c ng n a ana G a n*
n n b c a ng Han Sc D r a on D on o In or a c
n o n rg 00 //www / r / Sc
n a o R r ora an o Ran ng anar Gra
on g ra on or a r ng r an In *n n n b c an*
a n 00
ora J San an o on a co a on o r ng r
gra on n nown anar o c In *S n n n n g n b*
an Sy ag 00
ora J San o an agg n x r n n con ra n ng
on a ng r con ac c on w gr r g o r In *S n*
n n n g n b an Sy ag 00
R r an ra n r c a a o or a r ac ro aga on
arn ng R R a gor In *c ng n n n a*

Self-Organizing Maps versus Growing Neural Gas in a Robotic Application

Paola Baldassarri[1], Paolo Puliti[1], Anna Montesanto[1], Guido Tascini[1]

[1] University of Ancona, Computer Science Institute, via Brecce Bianche, 60131 Ancona, Italy
puliti@inform.unian.it

Abstract. The paper proposes a method for visual based self-localisation of a mobile agent in indoor environment. The images acquired by the camera constitute an implicit topological representation of the environment. The environment is a priori unknown and so the implemented architecture is entirely unsupervised. To compare the performance of some self-organising neural networks, a similar neural network architecture of both Self-Organizing Map (SOM) and Growing Neural Gas (GNG) has been realized. Extensive simulations are provided to characterise the effectiveness of the GNG model in recognition speed, classification tasks and in particular topology preserving as compared to the SOM model. This behaviour depends on the following fact: a network (GNG) that adds nodes into map space can approximate the input space more accurately than a network with a predefined structure and size (SOM). The work shows that the GNG network is able to correctly reconstruct the environment topological map.

1 Introduction

The preservation of neighbourhood relations, better known as topology preservation, is a very useful property of self-organizing networks and has attracted a great deal of interest [5], [8], [11]. A mapping preserves neighbourhood relations if nearby points in input space remain close in the map space. In general, a network can only performs a perfectly topology preserving mapping if the dimensionality of the map space reflects the dimensionality of the input space [7].
Then, the self-organising networks are usually considered as topology preserving models, since it is a consequence of the competitive learning: similar patterns are mapped into adjacent neurons and neighbouring neurons activate or code similar patterns (topology preserving feature maps) [2]. However it is not true in a great number of cases. Martinetz and Schulten [9] have formally defined what are Topology Representing Networks and its relationship with computational geometry structures as the Voronoi Diagram and the Delaunay triangulation. Then, several models are not preserving topology networks, for instance, Self-Organizing Map (SOM) of Kohonen [6], with a fixed dimensionality.
Then the self-organising network that better preserve topology is the Growing Neural Gas (GNG) of Fritzke [3], with variable dimension. In this case, the topology is not previously defined but it is adapted during the learning phase.

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 201-208, 2003.

In this paper we describe our application on robot self-localisation using images from a digital camera. So, to verify the performances in robotic field of some self-organising network models, we realise a similar architecture by using: in a first case the SOM of Kohonen; in a second case the GNG of Fritzke. Particularly, we compare the results considering the two different models, making a point of the preserving topology property that characterises the networks with variable dimensionality. The result is that the GNG network is better than the SOM network for representing the environment topological map. Then, similar input patterns from an input space are projected into nodes that are close to each other in the output map, and conversely, nodes that are adjacent in the output map decode similar input patterns [10].
We use the images as an implicit representation of the environment. The topological map consists in a set of images connected by links on a graph. Consequently the environment is characterised by a topological map that is used to qualitatively detect the position of the robot. The learning algorithm of proposed architecture detects the occurrence of unknown data and automatically adapts the structure of the network to the learning of these new data.

2 Introduction to the neural networks used

Self-Organizing networks are normally considered as Topology preserving of an input space and this is viewed as a consequence of the competitive learning. However, by following recent definition of topology preservation, we can say that not every self-organising model has this quality [9].

2.1 The Self-Organising Map (SOM)

The self-organising map of Kohonen [6] is one of the best-known and most popular neural networks in the literature. The SOM consists of a set of competitive units that form a lattice with a topological neighbourhood function. The algorithm itself, which is inspired by sensory mappings found in the brain, is relatively simple. The lattice of map nodes are each connected via adaptable weight vectors to each element of the input vector. The SOM algorithm performs competitive learning, but instead of just the winning node being adapted, nodes that are close to the winner in the map space (neighbours) are also adapted, although to a lesser extent.
This means that nodes that are close together in the lattice respond to similar inputs, so that the set of local neighbourhoods self-organises to produce a global ordering. The dimension of the map space and the number of nodes in the network are chosen in advance, and typically the map have one- or two-dimensions. In the one-dimensional case the map is organised to form a continuous ring of neurons. In the two-dimensional case the map is been bent to form a matrix of neurons.
For a given input, the distance between the input and each of nodes in the map field is calculated usually as Euclidean distance. The node with the minimum distance is selected as the winner and the weight for that node and its neighbours in the map field are updated. In the SOM the network structure and dimensionality must be decided

before the learning (fixed dimensionality). This constraints the resulting mappings and the accuracy of the output.
So, the SOM have only partially the topology preserving property: a neighbour neurons corresponds to neighbour input vectors, but no vice versa. A real topology preserving, may be obtained by models with variable dimension.

2.2 The Growing Neural Gas (GNG)

The GNG is an incremental neural network. The incremental character of this model, avoids the necessity to previously specify the network size [2]. It can be described as a graph consisting of nodes, each of which has an associated weight vector (reference vector), and by defining the node's position in the data space and a set of edges between the nodes and its neighbours. During the learning phase, new nodes are introduced into the network till a maximal number of nodes is achieved. The GNG starts with two nodes, randomly localized in the data space, connected by an edge. A new node is introduced into a network after a predefined number of iterations, λ (adaptation steps), defined by the user. The new node has to be slowly introduced into the network structure, in order to lieve enough time for redidtributing the nodes over the data space. For each data object the winner (the closest node) and the closest node of the winner, are determined. These two nodes are connected by an edge and with each edge an age variable is associated. At each learning step, the age of all edge emanating from the winner are increased by one. When the edge is created its age is set to zero. Edges exceeding a maximal age and any nodes having no emanating edge are removed. In this way the topology of the network is modified. The neighbourhood of the winner is limited to its topological neighbours, in other words, nodes connected by edge with a winner. In GNG all topological neighbours are updated in the same way, and the squared distance to the input data is accumulated for the winner [1].
This network is able to learn high dimensional input spaces, and to satisfy in every situation topology preserving property. This is because its topology is not previously defined, but they adapt it during the learning [2].

3 The implemented architecture

The realised system simulates the behaviour of a mobile robot, that is able to know and to classify the images acquired from a visual sensor, fixed on robot body. Our system uses a single sensor, that is an omni-directional digital camera that provides the description of the environment. This camera acquires the images with colour depth of 24 bits.
The choice of the environment to test the realised architecture is very important to verify the system effective ability. Then the choice environment is without chromatic characteristics and with not bright colours, a priori unknown, and in particular it consists in a typical indoor environment: two different corridors, where the system has to recognise the right zone of a specific corridor.

In the first time, we extract the frame from the videos relative to the two corridors. After we reduce the dimensions of the images through bilinear mean of the colours. The images are reduced in a bitmap format of 32×24 pixels and then every pixel of the images is separated into RGB (Red Green Blue) components. The components have a value between 0 and 255.
To verify the performances in robotic field of two self-organising network models, the following architecture has been realized using both the SOM and the GNG networks with the same structure. In all the examples, the output map of the SOM is a one-dimensional chain.

- The first layer consists of three neural networks, one for each component RGB, and it has to identify the space of the possible vectors, classifying the images according to levels of Red, Green or Blue about the individual pixels.
- The second layer consists of a single neural network. It receives in input the weight vectors of three winner units; one for each network relative to the first layer. Then, the layer simultaneously processes three colours levels. The output of the second layer is the winner unit and so the reply of the system.

In the second layer the nodes number is equal to the number of parts in which we divided the corridor. During the training phase, each node correspond to a determined part, depending on the resemblance to the training set data. The network also establishes the length of the corridor parts associated to the same node.
Then the network receives again in input the same training set and now the system determines the images group that has the same output, including uncertain zones with heuristic methods. In this way the system discriminates the images belonging to a specific part of the corridor and then it explicates the correspondence nodes-parts.
The training set used for the learning consists of the 80% of the images about the different locations that the system had to identify. To verify the real performance of the recognition algorithm, we uses the other 20% of images for the test.
During the test, the system associated each input image to an appropriate winner unit. In other word, the system correctly replies if the output of the system is the node that identifies the corridor part from which the image is previously acquired.

4 Experimental results

The behaviour of the implemented model is checked with some simulation's. In order to study the image recognition capability and the topology preservation capability, we have chosen networks with different characteristic, according to the number of units during the learning process and to the characteristics of the input. We have tested our method on real data images obtained from a digital camera, mounted on the top of the mobile platform. The environment consisted in two corridors, everyone with different characteristics. For the two corridors we have performed the learning of different input manifold, according to the images dimension and to the nodes number. The following presentation considers the best case for each corridor.

4.1 First corridor

The length of the first corridor is approximately 36 meters. During the run 1000 video sequences are taken, corresponding to 1000 different pose. The training set consists of 800 images, whereas the test set consists of 200 images. The images are rather similar and they do not show particular bright colours.
To evaluate the performances of the two self-organising networks we have chosen the same number of nodes in the network of the second layer. The number of nodes is 18. This choice depends by the length of the corridor and so the subdivision of the corridor in 18 parts there is. The time necessary to recognition each image is 0.065 seconds, using the SOM. The time is considerably reduced if we consider the same corridor using the GNG network. In fact, the time to test the architecture using GNG network is 0.037 seconds for each image.
The figure 1 compares the percentage of acknowledgement of the test images for each part of the corridor. We can see that the percentage of recognition has high value in either case, so both models are able to identify the images in a given distribution. The GNG model, however, in each zone has the best percentage.
So, the percentage of the acknowledgement of the all data set, during the test is 93,5% for the recognition using the SOM. While using the GNG, the percentage is 99,5%. In a global point of view, the GNG model acknowledges all the test images better than the SOM, although either value are rather high.

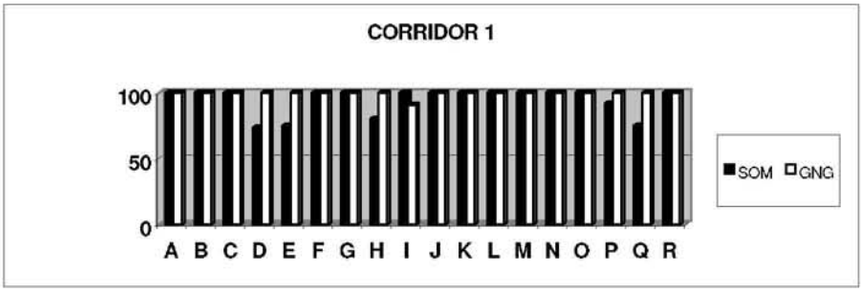


Fig. 1. Comparison of the percentage of recognition in each part between SOM and GNG

Another important aspect that concerns the topology preserving property is considered. So as final result, we want to obtain that the edges between the nodes in the last layer matches the topological map of the corridor. In this way, we can reconstruct the topology of the environment through the reply of the network, since each node identifies a particular zone of the corridor.
The following figure 2 compares the graph in output and its connections in two architectures (one with SOM networks and the other with GNG networks). The corridor is effectively and correctly reconstructed if the letters in the figure are linked in alphabetical order.
The triangles identify the connections between the nodes of the adapted SOM network. In this case the topology of the corridor in the last part of the corridor is only respected. In fact, in the initial part the nodes connect zones not nearby.

The squares identify the topological representation of the corridor, considering the GNG networks. We can see that the system now is able to connect correctly the neighbouring parts of the corridor, and then the preserving topology property is rather well respected. The stars identify some mistake, when the arcs of the graph also connect not neighbour parts of the corridor.

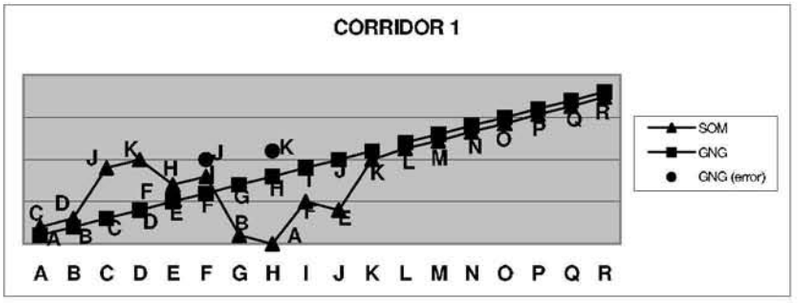


Fig. 2. Comparison on the topology preservation between SOM and GNG

4.2 Second corridor

The length of the second corridor is approximately 15 meters. During the run 900 video sequences are taken corresponding to 900 different pose. The training set is created from 750 images while the system testing is carried out on 150 different images.

The images of the last environment are more coloured and clearer than the others. Since this corridor is the most short, in the network of the second layer of the architecture, for either self-organising networks, we have considered 10 nodes. So the corridor is divided in 10 parts, in each zone the images are rather similar.

The time necessary to acknowledge each image is 0.048 seconds, using the SOM. While the time to test the identical architecture using GNG network is 0.035 seconds for each image.

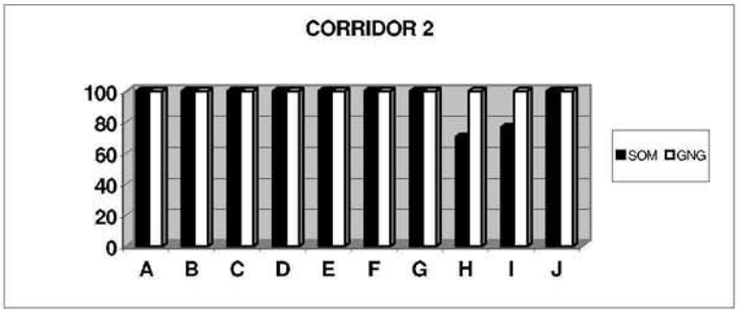


Fig. 3. Comparison of the percentage of recognition in each part between SOM and GNG

In the figure 3 we compare the percentage of recognition of the test images in each part of the corridor. We can see that the percentage of acknowledgement is 100% in each part, when in the architecture we consider the GNG networks. And so in this case we have a global recognition of 100%. Loosely speaking all the images are perfectly clustered and recognized. If we use SOM networks we have a smallest percentage to correctly classify the images. The total percentage is 90.6%. The GNG network has the best percentage, whether locally or globally.

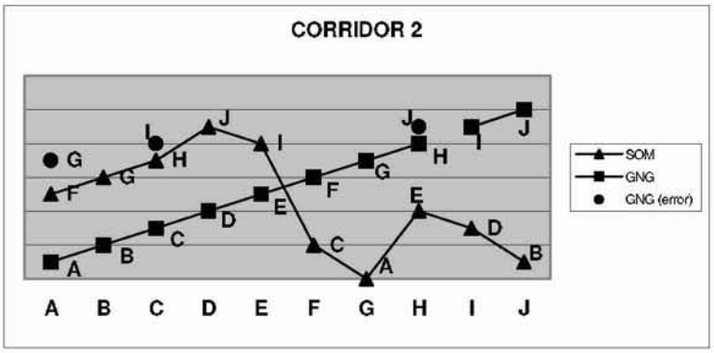


Fig. 4. Comparison on the topology preservation between SOM and GNG

By observing the figure 4, we can see that the connections between the nodes of the second layer network, respect rather well the topology of the second corridor, when the architecture consists of GNG network (squares). In this case the system acknowledges all the test images, and it is able to respect neighbourhood relationships in the zones of the corridor (figure 4). Then, the corridor is completely reconstructed and the topology is preserved. The arcs connect correctly neighbouring parts in the topological map of the corridor. The stars identify some erroneous connections for this particular situation.
The triangles represent the output nodes of the system when we consider the SOM model. As we can see the parts of the corridor are not very well connected, the letters in alphabetic order are not linked. In this experiment, it is not possible to recognize the topological map of the analysed environment. The links between the nodes do not reflect the connections between the adjacent parts of corridor.

5 Conclusion

The aim of this paper was to present an application of the topology preserving capabilities of two different self-organising networks. We have shown that the network with a variable topology (GNG) adapts better to any one of the inputs manifolds than network with a predetermined topology (SOM). We have presented a preliminary work on the use of artificial vision in a self-localisation schema for a mobile robot. The proposed approach, well suited for an office like environments, used an inexpensive camera with standard hardware. To compare the effectiveness of

the two models, an unsupervised architecture has been realized using both the SOM and the GNG, which have the same structure.
Experimental results are presented to show a good classification accuracy of the proposed architecture, in either case. In fact, both the SOM and GNG have been particularly successful in data recognition tasks, although the GNG network assures the best results and in a few time.
The simulations show that the pre-defined output map structure of SOM loose the topology preserving property; in fact the connections between the nodes of the adapted network does not reflect the sequence of different zones in which the corridor is divided. While, the GNG algorithm adjusts to the topological structure of a given input manifold and forms, always a perfectly topology preserving mapping. In fact, in this case the system not only optimally recognizes the images in each corridor, but it is able to reconstruct the topological map of each environment. In other words, the nodes and their connections reconstruct the corridor, since there is a correspondence "nodes-pieces". This property is very important for the blind navigation. When a mobile agent navigates in a unknown environment, the proposed architecture allows to automatic represent the topological map of the environment and so to support the moving agent. So, while the agent navigates in the blind environment it can perform an assessment of his localisation.

References

1. M. Daszykowski, B. Walczak and D.L. Massart, "On the Optimal Partitioning of Data with k-Means, Growing K-Means, Neural Gas", J. Chem. Inf. Comput. Sci., 42 (2002) 1378-1389.
2. F. Florez, J.M. Garcia, J. Garcia, A. Hernàndez, "Representing 2D objects. Comparison of several self-organizing networks", 3th WSES Conference on Neural Networks and Applications, Interlaken (2002).
3. B. Fritzke, "A Growing Neural Gas Networks Learns Topologies", Advances in Neural Information Processing Systems, 7, Brussels (1995), 625-632
4. B. Fritzke, "Growing Self-organizing Networks-Why?", European Symposium on Artificial Neural Network, (1996), 61-72.
5. G.J. Goodhill and T.J. Sejnowski, "A Unifying Objective Function for Topographic Mapping", Neural Computation, vol. 9, (1997) 1291-1304.
6. T. Kohonen, "Self-Organizing Maps", Springer Verlag, (1995).
7. S. Marsland, J. Shapiro and U. Nehmzow, "A Self-Organizing Network That Grows When Required", Neural Networks, vol. 15, (2002), 1041-1058.
8. T. Martinetz, "Competitive Hebbian Learning Rules Forms Perfectly Topology Preserving Maps", In the Proceeding of International Conference on Artificial Neural Networks (ICANN '93), (1993), 426-438.
9. T. Martinetz and K. Schulten, "Topology Representing Networks", Neural Networks, vol. 7, No. 3, (1994), 507-522.
10. J. Si, S. Lin and M.A. Vuong, "Dynamic Topology Representing Networks", Neural Networks, vol. 13, (2000), 617-627.
11. T. Vilmann, R. Der, M. Herrmann and T.M. Martinetz, "Topology Preservation in Self-Organizing Maps: Exact Definition and Measurement", IEEE Transactions on Neural Networks, vol. 8, No. 2, (March 1997), 256-266.

Towards reactive navigation and attention skills for 3D intelligent characters *

Miguel Lozano[1], Francisco Grimaldo[2], and Javier Villaplana[3]

[1] Computer Science Department, University of Valencia,
Dr Moliner 50, (Burjassot) Valencia, Spain
{Miguel.Lozano}@uv.es
[2] Institute of Robotics, University of Valencia,
Pol. de la Coma s/n (Paterna) Valencia, Spain
{Fran.Grimaldo}@irobot.uv.es
[3] Neurotechnology, Control and Robotics Group
Engineering and Automatic Systems Department
Politechnics University of Cartagena, Spain
{Javi.Molina}@upct.es

Abstract. This paper presents a neural design which is able to provide the necessary reactive navigation and attention skills for 3D embodied agents (virtual humanoids or characters). Based on Grossberg's neural model of conditioning [6], as recently implemented by Chang and Gaudiando [7], and according to the Adaptative Resonance Theory (ART) and the neuroscientific concepts associated, the neural design introduced has been divided in two main phases. Firstly, an environment-categorization phase, where an on-line pattern recognition and categorization of the current agent sensory input data is carried out by a self organizing neural network, which will finally provide the agent's short term memory layer(STM). Secondly, and based on the classical conditioning paradigm, the model will associate the interesting STM states, from the navigation or attention points of view, to finally simulate these necessary skills for 3D characters or humanoids. Finally, we will show some experimental navigational results, through the integration of the model presented in 3D virtual environments.

1 Introduction

Intelligent Robotics and Intelligent Virtual Environments(IVE) share the lack of designing agents capable of finding paths free of obstacles in order to satisfy their high level goals. Furthermore, one of the main tasks of any mobil agent, including humans, is navigation. In the majority of applications where 3D embodied agents are required, such as games or real time graphic simulations, this navigation problem is normally solved using any plausible global search technique, which is normally applied under classical static environmental assumptions [9]. A more reduced set of contributions in this field considers navigation as a local agent

* partially supported by the GVA-project CTIDIB-2002-182 (Spain).

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 209-216, 2003.

problem, simulating a realistic information flow from the environment to the agent, and easily computed through the classical *sense/think/act* agent cycle.

From robotics, Latombe et al. have combined 2D path-planning techniques with a rule-based system for modelling memory and synthetic vision/perception in virtual human simulations[4].

Tu and Terzopoulos implemented a realistic simulation of autonomous artificial fish combining a raycasting vision system with physical-based locomotion [5]. This system has been also applied to 3D virtual humans [12].

Noser et al. presented a synthetic vision system that uses object-false coloring and dynamic octrees to represent the visual memory of the character who navigates combining global and local techniques[2].

As nowadays, the increasing graphic realism of 3D characters has generated corresponding expectations in terms of character's behaviour (including the low-level ones), our main research interest is to provide for the adequated AI formalisms in order to reproduce and display the neccesary intelligence related to the skills cited.

In 3D environments, agents can perfectly access to the complete visual data base, so local sensorization is not really a requirement. However, local methods are clearly more appropriate for managing the necessary information flow to simulate reactive behaviours, such as navigation in dynamic environments or visual attention. For instance, the process of filtering visual sensory information and selecting the most interest objects from the environment to attend, that is, visual attention would be a globally untractable problem for complex environments. Furthermore, global navigation techniques for 3D intelligent virtual agents (IVAs) present the following problems:

- Reactivity: During navigation, the agent's target could change, for instance it could detect a virtual friend, or escape from a new enemy. Dynamic virtual environment simulations, where obstacles or agents could appear anytime, suggest the avoidance of global path calculations.
- Uncertainty: The agents could not have global information about their environment. This must be a requirement in virtual humans, which must only remember the places previously visited, in order to perform realistic simulations, such as in supermarkets, etc.
- Realism: Depending on the environment discretization (2D-grid, quadtrees, octrees) the global path obtained will contain the set of cell-centroids, which the agent must visit to finally achieve its position goal. The visualization of this path will have a low realism degree and normally an extra path-smooth phase will be also required.

Bearing this in mind, and although low-level navigation and attention behaviours for 3D characters depend on the specific application, we can identify these essential requirements:

- Reachability: For any goal position, the agent must be able to reach it autonomously.

- Collision Avoidance: Agents must reach their goal positions without colliding with any obstacle or agent.
- Replanning: Goal position can change during navigation so flexible navigation will be also required.
- Lifelike paths: Optimization criteria considered typically in global navigation algorithms, (minimum distance/energy), must be blended with local information in order to achieve lifelike paths instead of minimum ones.
- Lifelike attention: In every simulation cycle the agent must attend to interesting objects/agents, according to their properties (ej.: type, size, color, speed, direction, ...) .

The rest of this paper is organized as follows, firstly we will present the sensory agent system and the feedforward scheme introduced as the basis of the neural design proposed. Then we will explain this design focussing mainly on, the on-line environment categorization performed by a FuzzyArt Neural Network, and the conditioning machanisms introduced later. Finally we will also show the experimental navigation results obtained during simulation time, which have demonstrated a good potential for scaling the neural model up.

2 The sensory agent system

Intelligent robots or virtual characters with a set of sensors and memory skills must be able to explore unknown environments, and incrementally build their own internal model of the world. In this way, the main information channel between the environment and the agent, should be provided by the sensory agent system.Furthermore, 3D humanoids must take this into account, simulating a realistic information flow from the environment to the agent, instead of creating omniscient agents.

As we know, synthetic vision differs from vision computations for real robots, since we can skip all of the problems of distance detection, pattern recognition, and noisy images [4]. This allow us to implement a reasonable model of visual information flow that operates in real time systems. We are considering virtual vision sensors as a simple pyramidical culling volume from the agent point of view. According to this, in each agent's simulation cycle, every object or agent accepted by its vision cull-test, that is, the visible objects and agents, will shape an observation vector within semantic information relating to the object/agent (direction, distance, size, color, ...). In order to introduce high level information to be used by the planner system [11], we have also included relevant information about its state, for instance `door(open/closed)`(Figure 1). This semantic-vector approach is similar to Latombe's perception-based navigation system, where visual observations are simulated using an output vector provided by the vision module as well [4]. As Figure 1 shows, each observation corresponds to a couple of vectors, that is, the rays that cover completely the obstacle from the agent's point of view. This information will be useful for navigation tasks in two ways, firstly as the necessary patterns to represent and categorize the environment,

in order to perceive and detect them in the future, and secondly to inform the navigation system about the directions that will drive the agent to a collision situation.

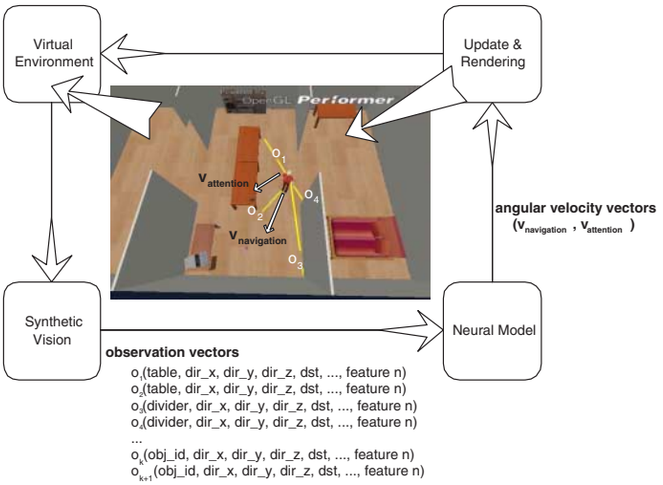


Fig. 1. Basic agent loop

In the next section we will review the neural model we are using to categorize the environment, as a generic machanism for the agent's perception and situation detection.

3 The Neural Design

The neural design proposed is based on a simple 3-layer feedforward model presented in [10]. The first layer (goal following) will compute the alignment of each neuron, as a vector of an angular map, to the goal. Then, in the correction layer, and according to the sensory information provided, we can reprimand the neurons which would locate the agent on a collision course. Finally both contributions are taken into account in the motor layer, which will search for the neuron with the maximum activation value. As we will see in the results section, this reactive scheme can be adequate for 3D virtual humans, and let us consider the possibility of parametrizing several important factors, such as, the security distance to objects. However, it could fail in some situations (Figure 4), local minima, etc, so we have complete the correction layer with the following model.

3.1 Synthetic Perception and Short Term Memory

Human vision is a complex process where input data is taken from saccadic eye movements and then processed in different areas of visual and prefrontal cortex. During this process a high level classification occurs resulting in a number of categories which will represent the state of the agent's environment, from its point of view. The Adaptative Resonance Theory (ART) has been used for this task as a well known human cognitive information processing model and according to this model, each agent will have a Fuzzy ART Neural Network for managing this categorization, which will give the agent the possibility of learning new categories, or situations to consider, without forgetting familiar ones [6].

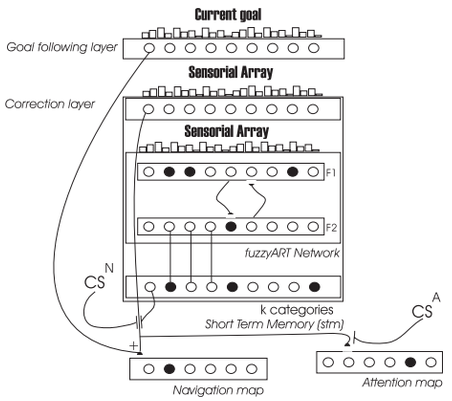


Fig. 2. Neural design for agent attentive navigation

As Carpenter&Grossberg pointed out, the Fuzzy-ART system tries to allocate the current sensory input sample in one of the familiar categories previously learned. When agreeing with the vigilance parameter, typically included in ART systems, this sample cannot be committed to any of the current categories, the model will create a new one. In this first learning step, the observation vectors (Figure 2) will be sent to the agent which will *take a look* at its environment for ascertaining its variability. As a self organizing neural network, this learning step can be carried out on-line, that is, in real time. Bearing this in mind, the agent will learn the variability of objects from its environment to finally manage a finite number of self developed categories which will represent several situations that the agent is interested in controlling, such us, a new interesting objects/ agents to attend, or a plausible collision situation.

Typically, in ART systems a single input pattern is able to activate only one category, however, a complete reasoning cycle for 3D IVA's must imply the sequential activation of different categories. According to this, the active categories will be stored in a new layer (STM) which will finally represent the agent's short term memory in real time (Figure 2).

The next problem will consist in finding the right associations between specific STM states and the 3D agent motor layers, considered as normal angular velocity vector maps. To achieve this, the model must learn *when* and *how* to map its current STM state into the navigation and attention motor layers. This process will be based on the classical conditioning paradigm and it will also be neccesary to consider specific conditioning learning signals in both associations: STM - navigation layer (CSn) and STM - attention layer(CSa) [7](Figure 2).

In a first mapping (STM-navigation association), the model will help the correction layer of the basic feedforward scheme to finally control the angular velocity vector of the agent's body, so that, the training will be oriented to detect and associate collision situations with the navigation output vector.

In parallel (currently under construction), a second mapping will simulate visual human attention. Again, associating the right stimuli, the velocity vector map considered for the attention layer will control the final character's head orientation in real time. For a further explanation about the neural model presented see [10].

4 Experimental Results

The model described has been implmented in C++ and has been integrated with two kinds of 3D environments, such as, SGI-IrisPerformer API and the Unreal Tournament 3D virtual environment graphic engine. The first results presented in [10] was concerned with the basic feedforward model, and no learning was carried out. This system has been upgraded and tested in several situations, as Figure 3 shows, where through classical parametrization, this basic model let's us consider the possibility to include internal agent variables, such as stress, or a security factor, in order to help in avoiding any obstacle (Figure 3).

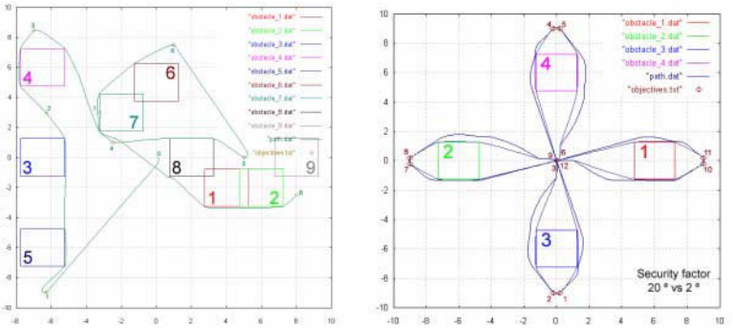


Fig. 3. Paths obtained in several test situations

Figure 4 in its first loop, shows the behaviour founded without considering the security factor mentioned, in this situation, when the agent is to close to the

bounding box of the object represented and no categories has been learned, it can loose its sensory information and go forward its goal, unfortunately, passing over the obstacle. As the categorization performed in the STM layer is sensitive to conflict situations, this let's us implement the associative learning already explained, in order to control them properly. This is shown in the second loop of its 7-step plan when a similar collision situation is faced the second time, the agent can now recognize it and correct it, avoiding the collision situation previously learned.

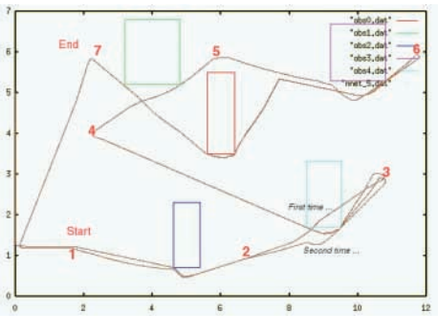


Fig. 4. Learning on-line from past situations

In order to visualize the resultant character's behaviours, the system has been also integrated in the Unreal Tournament 3D graphics engine, where basically the neural model should send via UDP sockets the current position and orientation to the game engine. These data are managed by several UnrealScripts, the engine's scripting lenguage, to finally locate the 3D character in the virtual environment (Figure 4).

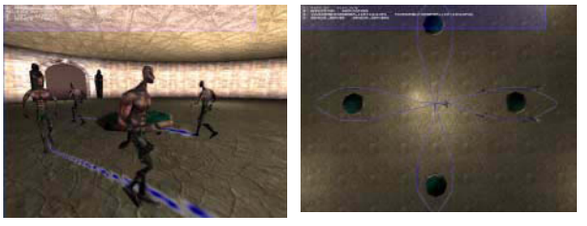


Fig. 5. Paths follwed by the 3D agent in a UT based 3D virtual environment

5 Conclusions

Neural approaches have been previously introduced in 3D intelligent creatures, mainly on solving dynamical systems for computer animations [5]. However, the majority of 3D agent architectures are focused on low cost global techniques to solve navigation problems and approaches from neuroscience are less frequent in 3D virtual worlds. A new neuromodel approach has been presented for covering visual attention and navigation for 3D intelligent virtual agents, such us 3D virtual humans but it could be also adapted to robotics.

According to the results obtained, we expect to undertake new experiments focussing on navigation reactions, for example avoiding dynamic obstacles or agents, and finally including the necessary machanisms for visual attention in 3D humanoids.

References

1. Aylett R. Luck M. -"Applying Artificial Intelligence to Virtual Reality: Intelligent Virtual Environments". Applied Artificial Intelligence, 2000.
2. H. Noser, O. Renault, D. Thalmann, N. Magnenat Thalmann, *Navigation for Digital Actors based on Synthetic Vision, Memory and Learning*, Computers and Graphics, Vol.19, No1, 1995, pp.7-19.
3. C. Elliot, J.Rickel J.Lester. "Lifelike Pedagogical Agents and affective computing: An exploratory synthesis". Artificial Intelligence Today (Editors: M.Wooldridge, M.Veloso), Springer 1999.
4. J. Kuffner and J.C. Latombe. " Fast Synthetic Vision, Memory, and Learning for Virtual Humans". Proc. of Computer Animation, IEEE, pp. 118-127, May 1999.
5. D. Terzopoulos. "Artificial life for computer graphics,". Communications of the ACM, 42(8), August, 1999, 32-42.
6. G. Carpenter, S. Grossberg. "A self organizing Neural Network for supervised learning, recognition and prediction". IEEE Communications Magazine, September 1992.
7. Chang C. and Gaudiano P. " Neural competitive maps for reactive and adaptive navigation". Proceedings of the 2nd International Conference on Computational Intelligence and Neuroscience, 19-23. Research Triangle Park, North Carolina, March 1997.
8. N. Badler and J.M. Allbeck *Embodied autonomous agents.* In Handbook of Virtual Environments, K. Stanney, Ed., Lawrence Erlbaum Associates, 2002, pp. 313-332.
9. M. Cavazza, S. Bandi, I. Palmer *Situated AI in Computer Games: Integrating NLP, pathplanning and 3D animation.*, AAAI Spring Symposium on AI and Computer Games, 1999.
10. M. Lozano, J. Molina. *A neural approach to an attentive nagigation for 3D Intelligent Virtual Agents.* IEEE System Man and Cybernetics October 2002, Tunisia.
11. M. Lozano, Mead, S.J., Cavazza, M. and Charles, F. *Search Based Planning: A Model for Character Behaviour.* Proceedings of the 3rd on Intelligent Games and Simulation, GameOn-2002, London, UK, November 2002.
12. T.F. Rabie D. Terzopoulos. "Active perception in virtual humans,". Proc. of the Vision Interface (VI'2000), Montral, Quebec, Canada .

Be e e
edge

p t d z d d z d d l

n r a ar o III a r a a n r a 0 gan
a r S a n
{ e e e e i u e

A st t N ra n wor a ro n o r ow r c n q
or o ng a w rang o a ow r arn conc ar n
r a a or an So wor r o o a n o c o ro
n wor onc n wor a n ra n a ow ng o n r
an o an o c on r or r a ar c o r o
an n r an ng a n ro o a roac a n ra
n wor o a con n o rang o a o n o g a o c
c n q co o wor w con n o c a o
wo ar o n r an or an In wor w r n a
a a o o a n ra n wor a o r cr ng
o w a c or q an a on a roac a ow ng o a
o c o

n uc i n

l tw s s l t s l w d t s s s s
l ss t p d t pt z t t t s w w w t t t
s l t t l t w s t l s t p l
l t s l p p t t t l pp s
l t p t t w t s d ffi lt t t p t t
s lts d t t t l l s s t t ss t l d
wl d ts p ss l d w s t p t
l t t t s p d l d s pt t s p ss
st t l s ll t d l t p s t t ll ws t
s l s wl d t s t t t l s Sp ll
ls s t st d t t w s t t wl d d ls
s l p s t t s s s s s s l d st d d
s t s l z t t l st t t p p s
lt t t t s sts d l t s w t
s l p l s w t tp t t t sp s t s p ts S
t d ll t s s pp d t s l l ss t t s w l ss
st l t ss t t t tp t t l tw t st t
p t t tw
t l ss tp t s t s w l s l s p s d l
l t s l 6 t p ll w w t d s t l ss s s pp

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 217-224, 2003.

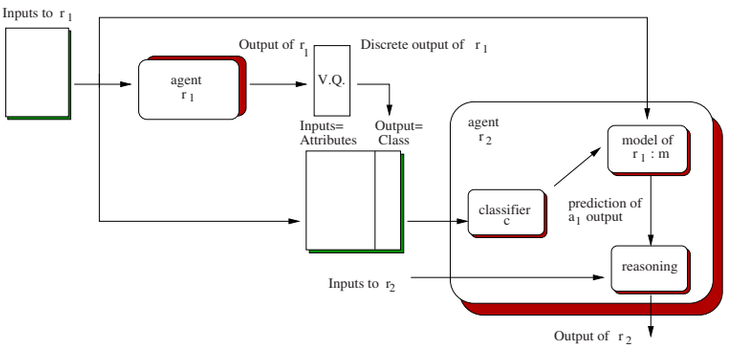
t t l s t t s l s t t d p l s ll s t s
d t s w w ppl t t z t t t l z d
l d l t L t d t s p d t t t z t s p ss t t
t s w ld t d d
t s p p S t p s ts l pp t s l d l
S t d s s t L l t S t d s s t w w
p ts w d d d p s ts t t d s lts ll S t
d s ss s t l s s d d t s

Au ic c uisi i n f s

t t t z d ts sp s t
p ts t t s s l t t t t s p ts S t s
s d d s l t s p tw p ts d tp ts t s
l ss t t s t s ll ws t d l ss p ss l tp t
t t s d l t d l t p s t t t
t sl t d t l ss t t s s s l t ds w
t l t s p tw t p ts d t tp ts st l t p
s w w p s t d s lts d s t t s s w t tp ts l d
t t s t l s w t ss pt t s t
t s s ls d s s t w w s t d d t ppl tw
d ff t pp s st t d s t z t tp t d d t s t p l
s l l ss l d s d t s s l l t t t
s l t d l w t t s tp ts l ss t s t p
p s t d s lts p s t d w t st pp
s s p l t d s t z t st p t t st d d
p l t s d pp s t t t tp t t ss t
s t s t s d t t p t t s t l t w t d l d
t d st d t t d t s w w ll w t
st pp t w l t t l w t d s t z t tp t s
pp s s d t t z t t ds p t l t L
t s w w s t t t ls t d t t t
t s l s s s t d l t tw t d l t t
l w s d s d w s ws t t l t
tw t t t d l $_2$t t t s t l d s t d l
t l ss t t *c* s d d l ts d t
t d l ss d l s l ss s ld d l t
t s w t t p s ts t s s t p t p tt s s
s d t t t d t tw t tp t p d d d
s ld l t t t z t st p s t d d
t d s t z t tp ts t s t d s s t s t

n i y A i A

l z d l d l t s l st t t t s sts
t t s p t t s t pp p t p t t s



F rc c r o o ng o ro o a or

t p t st t s t s d t t ds l t s s w t t s s p t s t T p t st t s d t s s tp t t s t w st t s u o l l tw d s d s s t d w s s t st s w t s t t l s t l st s t t t s l t s l d p d t t s d s w t t d w t pt lls t d

G n a z y g

g n w an n a co oo R a a G n a co oo o c r n r c n ro { r r ac c or a x n o on o c r n c ng on w o c n ro c o r o x n ar n g o r r R co c n ro or ac c r cr a R o o a n n w co oo ng q a on

$$n \quad R \quad \frac{\| \ \|}{R_j} \quad x_j$$

w r x_j R x_j a o co on n a r o c or x_j an R car na o R c I an c c r wa g n ra n r o an a rna co c or a gn n a n a o c n ro co a on o a rag or on or n or on a on c ang a a no g a o n nc a ra on

F 2 G n ra o gor

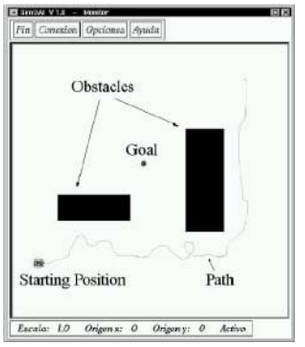
st p t t pt lls s l ss t t p s s l d s
l ll t l p t t p d s t s l d ff t t t s
pt lls l t d d pl d t s t t s lt spl tt
pt s ll t s s s pl d d pt

p i n up n su s

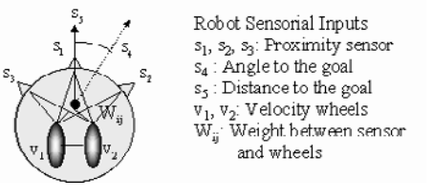
s s t d s s t p t l s d d t t s l
d l t w s s d d s l t t s d
t 2 d s w d t t p s s t t
p s t p s t d l t s d 2
d t t st d l t s l t

Tr

t t U l t w s s d t t t t
t ls t t t t s d t s
t s d s l w ld w t st l s w t d ff t
s p s ts l s t t l p s t ffi t w s s w
t s s s s s w t
t p t s s s t t w l s st l s t tw
s w t t t l t s d w t t l t t t l t s
t s tw w ls t t t d ff t sp ds v d v_2 s t
t t w p t l p p s t sp d w l v s d
t t l t l t w l v_2 t l tw
l d U l t s t t l t t s t p ts s s s t
v_2 sp d t d t d t ls t U l t



a n ron n w



Ro o rc c r

F Ro o an n ron n

The M e T

t t ll s l d t wl d t t t s t d l t
s t d l d l t ppl t l d l t
t d s t z t tp ts d t l d st ps t $_2$ s ll ws

t t ll d t l d l tw s
s l t s t st t t d s t s s s p ts
d t w l l t s tp ts c l d t p d t
t t t t t s t t s st t w d t
t s t pl s T s t t $_2$ l d d l
st t s t T s p s d t p t d t tp t
c t s t T w t t ll t c l s t t s t
T p ss l tp ts s s t s s d s p t t l z d l d
l t t t d d s t l s $\hat{T}$ st ll t c l s T
d s t z d s d $\hat{T}$ t $\hat{T}$
S t $\hat{T}$ s s d t t $_2$ w t s d l wl d s
wl d w ll t d s d s t t t t
w ll t s t l s t t d ls t t t s $_2$
wl d s ld p d t t tp t tt w t t s t t
t w tp t s w ls t st d d l s
st s

t s d t l t s s t p d t l ss s
s d tt t s s s s d t st s 6
sp d t s s l t s l ss s t d d
t st p t s l z t d st t t d t d s l t
d ff t l s
L s ls s d t t t l ss s s d ff t
d s t z t l ls 6 d 6 d t w l s l $_2$ wl
d d ls w t t ld ss l d t l s ss
t p t d $_2$ s s d s t pl s
w t p d t s $_2$ d ff t t t s t d s t z d l
p t t t t t t s t tp ts d s t z d l
l w t d l ss s t s t d L l ss t s lts
s z d t l t d s l t d t s s l t s
t d d ff t t s L d ff t d s t z t
l ls l ss s

roac	a	oc	L	L	L	L	L	L
cc rac		%	0%	0%	0%	0%	0%	0 0%

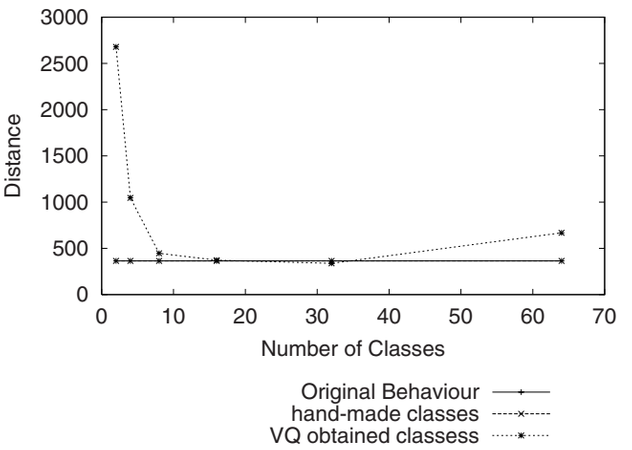
a ca on r

l t l ss s s s t d s s s s
s w l t l ss s t d ff t t s s t l ss t

p l s d t s l w w w ll s w l t t t t s s t p l
d t t d s t pl d d l

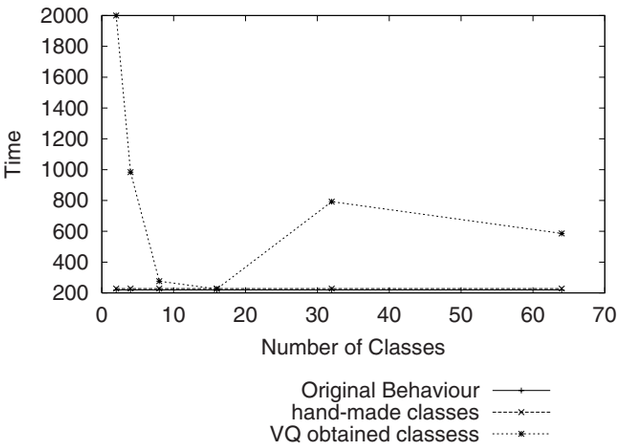
3 Te he e M e

st d t d ls t d d s l t d t 6 d ls w t
d ff t tp ts t d w t L t d w t st d
t t s l t S 6 d w p d t p
d l st t t t ll d t t s t
s d d d st d t t l t t l t w
t ls t t w t d t d ls t l s
d ff t pl s d s l w ld d s sts t l
ffi t w
s ws d st d d l l d t l
t t l t t ls w l s ws t t sp t
s d st l d l d t d l s d
l ss s t s l s lts s w t t t l st pp s s ss l



F o r anc

w s t d ls t d d s t z t l ss s w t L w
s t t w t l d d ff t l ss s s lts w s t p s
s l t s s w ll s w t l ss s s d 6 l ss s
t st s t s s t d ff t l ss s t s t t
t t t s l w s lt t d ll t s w s
s s l s s w l t l d d l s t d t s d
s d sp t t t t t t d l w t l ss s d spl s s l s lts
d st w t l d l t t s w s s s d



F 5 o r

t t s d l s t s t s t ls w t t s l d d d s t
l w d st t w t t t l t
t s t d 6 l ss s s d t d s t z t tw
tp t t d l t d s s l l t t s t d t
l d l d t d l w t l ss s d s t z d d

C nc usi ns

s lts s w t t pp s t d w d l s pl l
ts l d s l t l s st l d t
d t s p p w s d l t t
t t d l wl d t ppl d t l d l t t t t ll
t s ll d s t z d s t l ss s
d s t z t d t s d d w s l ss t t s t t
t s t l ss s d w t tp ts s l d
st d l s t s w w s d tw pp s t d t s
t t st s t d t d s t s t l ss s s t d
t t s l s s t d t l t t t t l ls t s
pl s w l st s w s ll d t d w t s p ss t
lw s d d st w d w t s l tp ts lt
d s l d t s ld t t d t t t d
t s w w s d t t z t t d t t p t s t
t z t l ls ll w t d s d l t t t s s lts s
l t t s lts t d t l d l d t d l d s d w t
l ss s p t d d w s d s t t s l s s
d t t t s pp
t t w w t t t d pp w t t ll w d s st
t t st pp w t t st l t s S d t s

ts l d ls t ts s t t t p t t t st
t s p ts t s t d ls t ts s ff l w
t w ld t st t t t ld l t t ts
s t l

f nc s

D R ar G n on an R J a arn ng r r n a on
ac ro aga ng rror *a* o
a ng c a an an n r w oor R n orc n
arn ng r *n* *c a n g nc a c*
n on o r anga rac Sanc ro I a an Jo o na g n ra
co o on o o g n ra a ono o ro o na ga on a or n
c ng ng n na y a n a Jo a San D go
S J 000 I r
J S a an G offr G ow *ac n a n ng a gy*
ac o I c a r R n ng S o c now g ng N ra N wor
0 organ a ann
R car o r Dan orra o In Ga an an ga o a arn ng
o o o r ag n n *c ng g n n*
a n ng g n arc ona S a n J n 000
J R n an In c on o c on r *ac n a n ng* o no
0
S o a q ar q an a on n *c* n a ora o
r c n ca No or on r n a In o a a ca S a c
ng an c N w J r S r n arc
c a on q an a on o I ran ac on on In or a on or

J Ro n an *g a ac n a n ng* organ a ann San
a o
r an J r an n an J S on *a ca n an*
g n a wor roo on r S
0 J Ro n an o n ng n anc a an o a arn ng n
c ng n n na na n nc n ac n a n ng r
J n organ a ann
n on o r anga ga o a an R car o r x rac ng now g ro
r ac ro o a o r n *c ng g n 1 n a n ng*
g n on r a ana a 00
n G r o an Ro r Gra *c an za n an S gna n*
w r ca c r
rnan o rnan an Dan orra o ng c or q an a on
o r n orc n arn ng n *b b S cc* n r
n c r No n r c a In g nc 0 S r ng r r ag 000
ra n rg *c n n yn c yc gy* I r a
ac
J Ro n an *g a ac n a n ng* organ a ann
So ar ga I r no a an an J o na r a
or or n g n a ono o ro o n *n cc ng n na na*
Sy n n g n b c Sy

f e e f d
e e y e e

d S S S ,2 st U 2 d

S gna roc ng Gro n r a c Sagra a a a 0 00 c S a n
i e u i e
a ora o r I ag S gna x NRS R *n* 0
IN G n x a 0 Gr no x ranc
i ti utte i g
a ora o r o a on a c 0 Gr no x ranc
i u i g

A st t a r ro o a r a o or n n a
ng a non n ar a ng w c ran or a o ran o ar a
o ro a r r ng goo a rox a on n w n
a c a on no a o can n cc
or n a ng non n ar n o non n ar x r or n n r
n r on or ro ng a gor an con rg nc

n uc i n

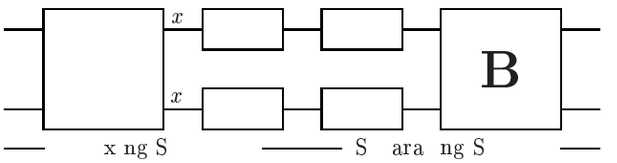
l d S p t d p d t s s SS s s p l s l p
ss w s s d d t s l t l st t s l
l st t s s w ll s l t t s tl w
s s dd ss d t p l s s p t l
t s w s s t s **e** *f* sp ll l d tt
st d d sp l d l st s l t s ll d p st l
t s w s p l s s w t s tw st
s st s sts l t ll w d p tw s l
d st t s t t p d s t s t s

f

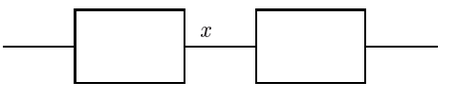
w t d p d t s s s t t s t d t s
t t s t w t d *f* s t w l
pp t p t

⋆ wor a n ar n ro an ro c In So rc S ara on
an a ca on ISS IS 0 D r cc o G n ra R c rca
a G n ra a a a n a n r a gran or In gra c on I 00 an
n r a c n r gran R0

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 225-232, 2003.



F x ng ara ng or N x r



F 2 n r con o a r o ow a or on

p l l d s s st s s l t t
t s s p t p l t s ts tp t w t s s

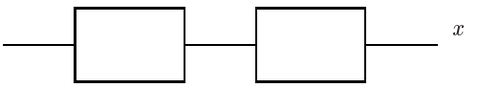
$$f$$

w s t d p d t d d t ll d st t d d p t s
t s t d t s t t s t w lt H d f s t
w l pp ss d t l d l ss
l d s p t s t d ls s st t st t
t s t l pp d t t s t l p t s
d z t t l t t s st t
tp t w t l ds t sl w d st s l t s s t tw
p ts s d d pt z d w t t s t
t s p p w p p s s pl st d ffi t t d l
st t t s t l pp S t pl s t p
pl s S t p p s s tw l ss s l t s d s t s ws s
p t l s lts w s t s t d t l z l d s
l t s d p ts sp d

incip s

The c

t d l s d t s l st t l pp
t t p t t t s l s
w t d s d l s d t t t l t
t ds t w d ss d l l pp f s
t d st t d s tl w ss t t t d l



F n r on

f s ss w p p s t st t t s
f s t l pp w s t d l
t ss
S l l t s st s t lt d s l
st t l t s w t d s d l s d
t t t l t t d l ss t d t t ds
t ss d l s t t t ss l
d p ds t lt t s l s t ss d st t t t d st
t t l s *S* t p p s t pp t t s
f t t s t t s ss
t t s t s t tw p l s s l w l t
d s pl t t t s

ve e y c

s pl st pp p t s s d t p p t t
l t d s t t s d t d l d d t ts
l t d s t t

w d t s t p l t
d l s t l d st t d
t t ss l t d s t t w t s s
t ss l d l t s l
t t s t ss d l
s pl pp t t s t l pp *f* s

^

3 M x z h e r y

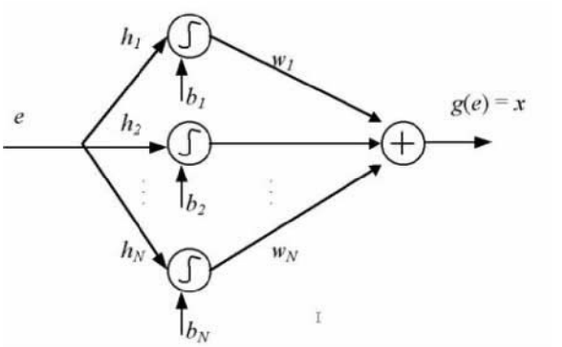
s d w t S t p t t d l

H *o* *z* *z*

w *z* d t s t p l t d s t t
S t d l t S t p *H* s
s ss t st t d s t t *H* *H*
s d t st t

A i s

Us t p s s lts p p s tw s pl l t s t
st t t s t l pp *f* st l t s
s s d t l d d S s t tl d
s s pl d st
s d l t s d t s lt S s t s sts d
st l pp s t t t S s t p s
d t st t [2] s s l w p t z
t l t pl s l tw s lt l
p pt s s w d



F a r rc ron

d t pp s l d d d st t p t s
h d t pt z t st t , ,**w** *H* lt
t s d d s st ll t s pl t l ds t l t w s
pl t d d l t t t p s
t ll w w l p t l s lts w t t s pl st
d ffi t l t s d

p i n su s

r c

d t t st t l t w d p ts s s s ls w t
d ff t t s s
l s d d t t l t s l
lt s pl t w l p t s lts w t l lt d
t t s l s lts t d w t

Ten different filters (low-pass, high-pass or band-pass) have been used for providing filtered signals with various distributions. Then, we apply the nonlinearity f. We check, as expected by theory, that the algorithm is completely independent of f since, $\forall f$ the function $\Phi^{-1} \circ F_E \circ f$ transforms the random variable X to a Gaussian variable Z. If the compensation of the nonlinearity was perfect, the function $\Phi^{-1} \circ F_E \circ f$ should be the identity function. Of course, it can be rigorously true, only if X is Gaussian.

For testing the algorithm of PNL source separation, we did some experiments for mixtures of two uniformly distributed random sources. The method used is the algorithm proposed by Taleb and Jutten [8], denoted TJ. The mixing system is composed of:

$$\mathbf{A} = \begin{bmatrix} 1 & 0.4 \\ 0.7 & 1 \end{bmatrix} \tag{6}$$

$$f_1(x) = f_2(x) = 0.1x + \tanh(10x) \tag{7}$$

This mixture leads to the joint distribution showed in Fig. 5, where the effect of the nonlinearities is clearly visible (left). In the same figure we can see the scatter plot after the initialization of nonlinear functions g (center) and the scatter plot of the initialized outputs $\mathbf{y}$, where the signals are decorrelated (right). It is easy to see qualitatively the initialization provides an estimation $y(t)$ which is a mixture, simpler than $e(t)$.

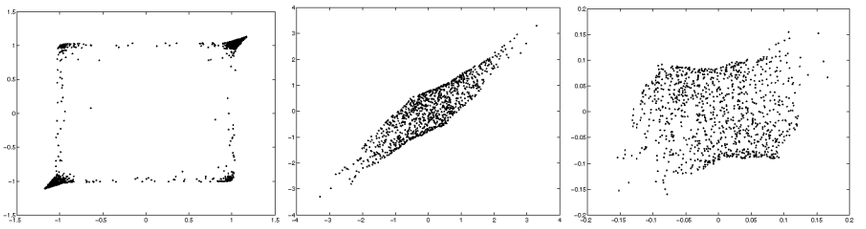


Fig. 5. Scatter plot of the observed signals (left), of the signals after initializing nonlinear functions g (center) and of the decorrelated output signals (right).

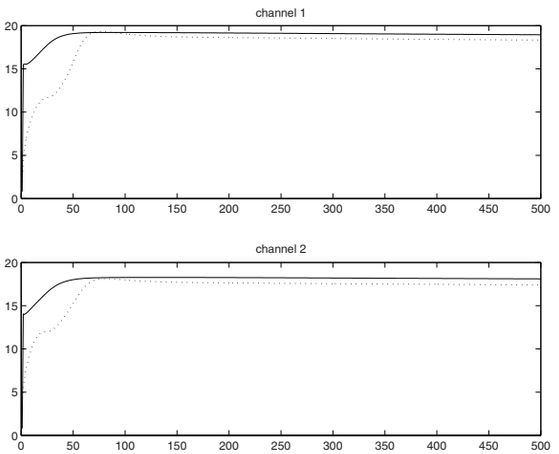
Despite hard nonlinearities ($0.05x + tanh(10x)$) are used in the experiments, the results obtained with the TJ method are satisfactory. The initialization process increases the convergence speed of the algorithm, and sometimes gives a better result in terms of output SNR (Fig. 6).

4.2 Results

The performance index $\boldsymbol{\varepsilon}$ of the compensation will be simply the mean square error[4]:

$$\boldsymbol{\varepsilon} = E_X[\|(\Phi^{-1} \circ F_E \circ f)(X) - X\|^2] \tag{8}$$

[4] For computing $\boldsymbol{\varepsilon}$, $x(t)$ and $z(t) = (\Phi^{-1} \circ F_E \circ f)(x(t))$ have to be normalized



F SNR o on or J a gor w n a a on roc o n an w o a n

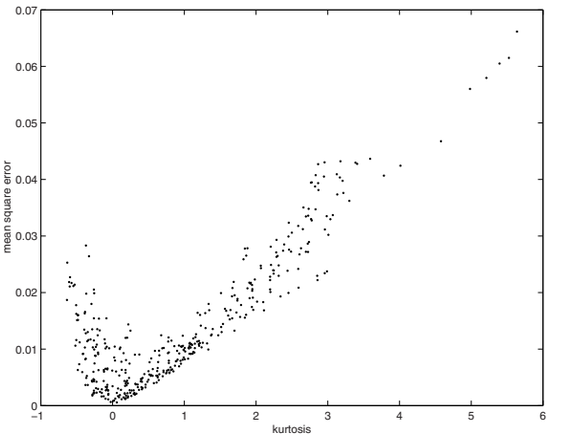
w s s t d tw *f* d t l t s ws t p d s s t t s s t lt d s ls t d w t t d ff t s s d lt s t t t p s w t s l t s s s l s t z s l s t ss d s s s t t s s s w z s s lt t ffi t t d s l l t d t d st t st t l t *f* l s t t ss t d st t l s t z t d w s w pl d d p p s t t l t d p s t sp ds t t s s d t p s t 6

C nc usi n

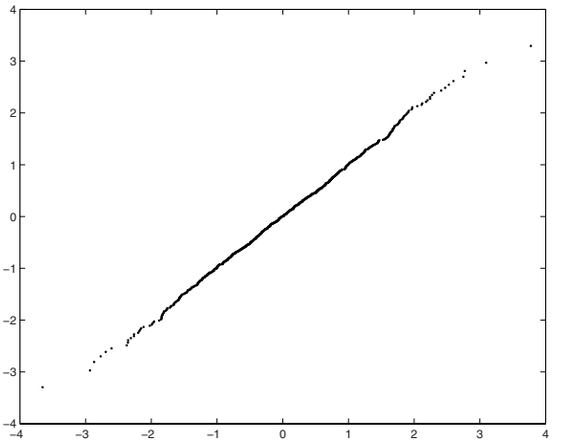
t s p p w p p s s pl d st t d l dl pp t l pp t d s s d t ss pt t t t p t l t l pp s ss d t t lt s lts s w t t d s st t t ss ss pt t w st s s t l ds t pp t t l pp w s d s t l z t l SS l t s

f nc s

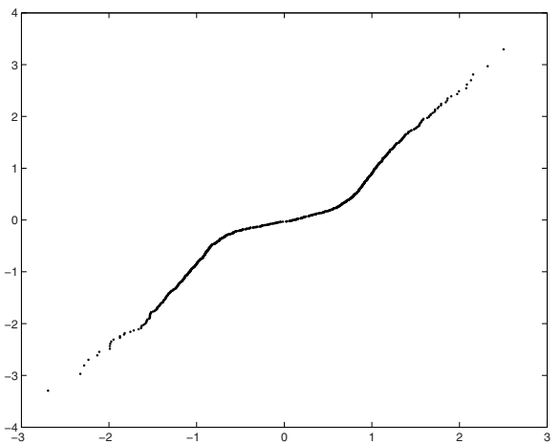
G r n ara on o o rc a non n ar n ra a gor *a* o



F r or anc n x r r o



F ca non n ar nc on co n a on



F or ca non n ar nc on co n a on

arra G D co an S ac S a ca n n anc an no
c on w n or a on r r ng non n ar a *a* *a* *n* o
0
a n n ar n n an J ar n n Non n ar o rc ara on
organ ng a n ong ong S r
a an J n n ro o a on a ca on o n o rc a
ra on n a ann Sw r an c o r
o r an R rg r n ara on o non n ar x ng
o n *S* or a S 0
G n on R ar an r o r o S ara on o o rc n
a c a o o non n ar x r n *S* *δ* r g g r
ang S ar an c oc In or a on or c a roac o n
ara on o o rc n non n ar x r *S gna* *c* *ng* o no
00
a an J n So rc ara on n o non n ar x r
n S gna *c* *ng* o no 0 0 0
G arq an a S ara on o non n ar x r ng a rn
r on n o R N
0 awana S ar ng an R r S ara on o o
non n ar x r ng ac an ora corr a on n *1* San
D go a orn a 00
a J So a a an J n a non ara r c n n r on o
n r *n S gna* *c* *ng* o no 00
o r an J o a *n* *n* *a* *n* *y* S r n
co n ca on
J So a a a a a J n an D a I ro ng a gor
n N x r ara on an n r n r on n *c* *ng*
Nara Ja an 00

Evolutionary Algorithm Using Mutual Information for Independent Component Analysis

F.Rojas, C. G. Puntonet, M.R.Álvarez, I.Rojas

Dpto. Arquitectura. y Tecnología de Computadores. University of Granada (Spain)
{fernando, carlos, mrodriguez, irojas}@atc.ugr.es

Abstract. Independent Component Analysis (ICA) is a method for finding underlying factors from multidimensional statistical data. ICA differs from other similar methods in that it looks for components that are both statistically independent and nongaussian. Blind Source Separation (BSS) consists in recovering unobserved signals from a known set of mixtures. Thus, ICA and BSS are equivalent when the mixture is assumed to be linear up to possible permutations and invertible scalings. However, when the mixing model is nonlinear, additional constraints are needed to assure that independent components correspond to the original signals. In this paper, we propose a simple though effective method based on estimating the probability densities of the outputs for solving the BSS problem in linear and nonlinear mixtures making use of genetic algorithms. A post-nonlinear mixture model is assumed so that the solution space in the nonlinear case is restricted to signals equivalent to the original ones.

1 Introduction

Independent Component Analysis (ICA) is a method for finding underlying factors or components from multidimensional or multivariate statistical data [6]. This technique deals with the problem of transforming a set of observation patterns x**,** whose components are not statistically independent from one another, into a set of patterns $y = F(x)$ whose components are statistically independent from one another. In linear ICA, which is the most extensively studied case, the transformation F is restricted to being linear. Nonlinear ICA allows F to be nonlinear. The principle that guides ICA is independence, i.e. the components y_i should be statistically independent, meaning that the value of any of the components gives no information on the values of the other components.

The majority of the methods for performing ICA use an objective function, also called contrast function [4], which measures the degree of dependence of the components of estimated sources. A contrast function is a mapping ψ from the set of probability densities $\left\{p_x, x \in \mathbb{R}^N\right\}$ *to* $\mathbb{R}$ satisfying the following requirements:

i. $\psi(p_x)$ does not change if the components of x_i are permuted.

ii. $\psi(p_x)$ is invariant to invertible scaling.

iii. If x has independent components, then $\psi(p_{Ax}) \leq \psi(p_x), \quad \forall A$ invertible.

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 233-240, 2003.

Thus, a good measure of dependence must be a function which has an absolute minimum which is reached only when the components are independent. When the mixture model is assumed to be linear (Linear ICA) the solution space is relatively constrained, therefore linear ICA methods do not need to be based on strict dependence measures. For example, some of these methods, which give rather good results in appropriate situations, are based only on cumulants up to the fourth order.

Nonlinear ICA, on the other hand, is rather unconstrained, and normally demands a good dependence measure. Various dependence measures have been proposed for this problem. Rojas et al [13] proposed a contrast function which approximate the mutual information of the estimated components. Mutual information gives a rigorous justification for the heuristic principle of nongaussianity and also serves as a unifying framework for many estimation principles, in particular maximum likelihood (ML) and maximization of nongaussianity.

Mutual information I between the elements of a multidimensional variable $\boldsymbol{y}$ is defined as:

$$I(y_1, y_2, ..., y_n) = \sum_{i=1}^{n} H(y_i) - H(y_1, y_2, ..., y_n). \quad \textbf{(1)}$$

The interpretation of mutual information is easily derived from Equation (1): how much information does one random variable tell about another one? Certainly, considering the case of two variables (y_1, y_2), the information that y_2 tell us about y_1 ($I(y_1, y_2)$) is the reduction in uncertainty about y_1 due to the knowledge of y_2 . In the case that all components y_1 ….. y_n are independent, the joint entropy is equal to the sum of the marginal entropies. Therefore, mutual information will be zero. In the rest of the cases (not independent components), the sum of marginal entropies will be higher than the joint entropy, leading thus to a positive value of mutual information.

Mutual information complies with the conditions given above for a contrast function: it does not change with permutations and scalings and reaches a minimum if and only if the components are independent. Unfortunately, to exactly compute entropies we need to know the analytical expression of the probability density function (PDF) which is generally not available in practical applications of BSS.

In this paper, we propose a simple approximation of mutual information based on estimating the PDF through discretization of the output space by histograms. The space of solutions will be explored through the use of a Genetic Algorithm, which it is also an important innovation in the field of ICA, due to the tendency of using adaptive approaches for this problem. The paper is organized as follows: linear and nonlinear mixing models for BSS will be shown in Section 2, together with their assumptions and restrictions. In Section 3, the genetic algorithm approach to BSS will be introduced, drawing special attention to the construction of the fitness function based on density estimation. Section 4 shows some experimental results over linear and nonlinear mixtures. Finally, a conclusion and remarks on future work finish the paper.

2 Blind Source Separation Mixing Models

The simplest BSS model assumes the existence of n independent signals $s_1,\ldots, s_n$ and the observation of $x_1,\ldots, x_n$ instantaneous linear mixtures (see Fig. 1a). This linear model is represented by the following equation:

$$x(t) = A \cdot s(t)\,. \tag{2}$$

where A is some unknown $n \times n$ matrix of real coefficients.

In the linear case we need to make the following tolerable assumptions, so the "blindness" of the method may be questioned [8]:

i. The sources are statistically independent of one another. This is the main assumption of most of the BSS algorithms.
ii. The matrix A is assumed to be invertible (i.e. non-singular)
iii. The sources have at most one Gaussian distribution.

Under this assumption we want to obtain a matrix W (separating matrix) whose output $y(t)$ would be an estimate of the sources $s(t)$:

$$y(t) = W \cdot x(t)\,. \tag{3}$$

Unfortunately, in many real world situations the linear assumption is an approximation of nonlinear phenomena [9]. For several situations, the linear assumption may lead to incorrect solutions. Hence, researchers in BSS have started addressing the nonlinear mixing models ([12], [14], [15]). A fundamental difficulty in nonlinear ICA is that is highly non-unique without some extra constraints, therefore finding independent components does not lead us necessarily to the original sources [7].

Source separation in the nonlinear case is, in general, impossible. Consequently, in this work, we assumed the so called *post-nonlinear model* [14] (see Fig.1b). The sources are first mixed linearly and, after that, a set of nonlinear distortions (f) is applied in each channel to get the final observations (x):

$$\begin{aligned} x(t) &= f(A \cdot s(t))\,. \\ y(t) &= W \cdot g(x(t))\,. \end{aligned} \tag{4}$$

For this model, the indeterminacies are the same as for the basic linear instantaneous mixing model: invertible scaling and permutation. The post-nonlinearity assumption is reasonable in many signal processing applications where the nonlinearities are introduced by sensors and preamplifiers.

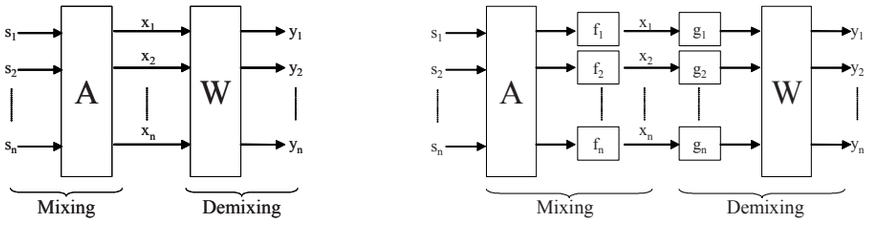


Fig. 1 Linear (1.a) and post-nonlinear (1.b) mixing and demixing models.

3 Genetic Algorithm to Approximate Mutual Information.

3.1 Entropy Approximation.

From Equation 1, we observed that minimizing mutual information was equivalent to obtaining estimations as independent as possible. The main difficulty was that in practical applications the probability density functions of the sources nor the estimations will be available. Thus, we propose to approximate densities through the discretization of the estimated signals building histograms and then calculating their joint and marginal entropies. In this way, we define a number of bins m that cover the selected estimation space and then we calculate how many points of the signal fall in each of the bins ($B_i \; i = 1,...,m$). Finally, we easily approximate marginal entropies using the following formula:

$$H(y) = -\sum_{i=1}^{n} p(y_i) \log_2 p(y_i) \approx -\sum_{j=1}^{m} \frac{\#B_j(y)}{n} \log_2 \frac{\#B_j(y)}{n}. \quad \textbf{(5)}$$

where $\#S$ denotes cardinality of set S, n is the number of points of estimation y, and B_j is the set of points which fall in the j^{th} bin.

The same method can be applied for computing the joint entropies of all the estimated signals:

$$H(y_1,...,y_p) = \sum_{i=1}^{p} H(y_i \,|\, y_{i-1},...,y_1) \approx -\sum_{i_1=1}^{m}\sum_{i_2=1}^{m}...\sum_{i_n=1}^{m} \frac{\#B_{i_1 i_2 ... i_p}(y)}{n} \log_2 \frac{\#B_{i_1 i_2 ... i_p}(y)}{n}. \quad \textbf{(6)}$$

where p is the number of components which need to be approximated.

Therefore, substituting entropies in Equation 1 by approximations of Equations 5 and 6, we obtain a contrast function which will reach its minimum value when the estimations are independent. The final step will be designing an algorithm that minimizes this contrast function, escaping from local minima. To this end, a well known paradigm is described in next section: genetic algorithms.

3.2 Genetic Algorithm.

Genetic Algorithms (GAs) are nowadays one of the most popular stochastic optimization techniques. They are inspired by the natural genetics and biological evolutionary process. The GA evaluates a population and generates a new one iteratively, with each successive population referred to as a generation. Given the current generation at iteration t, $G(t)$, the GA generates a new generation, $G(t+1)$, based on the previous generation, applying a set of genetic operations. The GA uses three basic operators to manipulate the genetic composition of a population: reproduction, crossover and mutation [5].

A genetic algorithm is characterized by the following features:

- A scheme for encoding solutions to a problem in the form of a chromosome (chromosomal representation).
- An initialization procedure for the population of chromosomes.
- A fitness or evaluation function which indicates the *fitness* or *aptitude* of each chromosome relative to the others in the current set of chromosomes (referred to as population).
- Genetic operators which are used to manipulate the composition of the population.
- A set of parameters that provide the initial settings for the algorithm: the population size and probabilities employed by the genetic operators.

Encoding Scheme.

i. Linear mixture: the genes of the chromosome will correspond to the coefficients of the separating matrix W (see Equation 3)

ii. PNL mixture: the genes will represent the coefficients of the odd polynomials which approximate the family of nonlinearities g (see Equation 4). The linear part of this model will be approximated by a well-known method such as JADE [3].

Initialization Procedure. Both polynomial and matrix coefficients which form part of the chromosome are randomly initialized.

Fitness Function. The key point in the performance of a GA is the definition of the fitness function. In this case, the fitness function that we want maximize will be precisely the inverse of the approximation of mutual information given in Equations 5 and 6:

$$Fitness(y) = \frac{1}{I(y)} = \frac{1}{\sum_{i=1}^{p} H(y_i) - H(y_1, y_2, ..., y_p)} \tag{7}$$

This fitness function will not change from the linear to the post-nonlinear case due to the fact that in both circumstances, the objective of finding independent components as equivalent to the desired sources does not change.

Genetic Operators. Typical crossover and mutation operators will be used for the manipulation of the current population in each iteration of the GA. The crossover operator is "Simple One-point Crossover". The mutation operator is "Non-Uniform Mutation" [10]. This operator presents the advantage, when compared to the classical uniform mutation operator, of performing less significant changes to the genes of the chromosome as the number of generations grows. This property makes the exploration-exploitation trade-off be more favorable to exploration in the early stages of the algorithm, while exploitation takes more importance when the solution given by the GA is closer to the optimal solution.

Parameter Set. Population size, number of generations, probability of mutation and crossover and other parameters relative to the genetic algorithm operation were chosen depending on the characteristics of the mixing problem.

4 Experimental Results.

To provide an experimental demonstration of the validity of the genetic algorithm to the BSS problem, we show here two simulations: one with a linear mixture and a second after a post-nonlinear mixture in a system of two sources. As measures of the accuracy of the methods, we applied the Normalized Root Mean Squared of Errors (relative to the standard deviation) and the Crosstalk (in decibels).

4.1 Linear mixture.

Two independent sources corresponding to a voice saying "*Adiós*" ("goodbye" in Spanish) and a telephone ring (10.000 samples) were mixed according to the following matrix:

$$A = \begin{bmatrix} 0.9 & 0.2 \\ -0.3 & 0.8 \end{bmatrix} \tag{8}$$

In Fig 2.(a), sources, mixtures and estimations are plotted. As can be seen, estimations are almost equivalent to the sources, up to scalings. The experiment was repeated in order to validate the stability of the algorithm, which is not deterministic. The results were, in general, of the same order as those shown here. In Fig. 2(b), the convergence of the algorithm is graphically showed, as the best individual in each generation does not improve considerably former ones from around iteration 30.

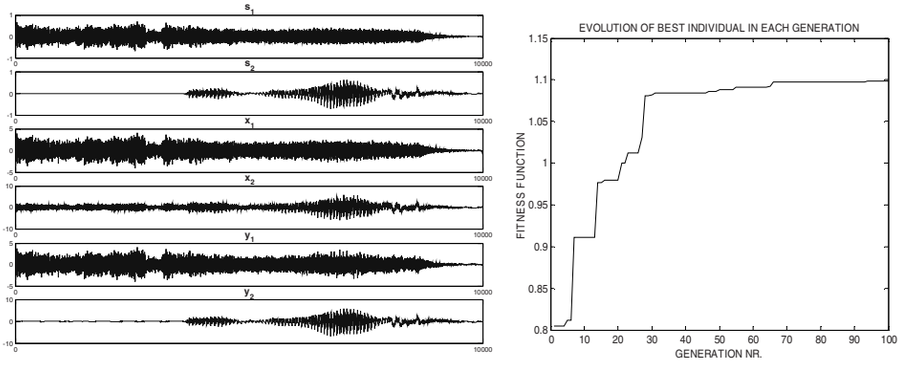


Fig. 2 (a) Plot of sources (top), mixtures (middle) and estimations (bottom) (b) Evolution of the best solution in each iteration

Crosstalk(y_1, s_1) (dB) = -27.78 dB, NormSQRMS(y_1, s_1) = 0.4083×10^{-3}
Crosstalk(y_2, s_2) (dB) = -75.09 dB, NormSQRMS(y_2, s_2) = 0.0018×10^{-3}

In this practical application, the population size was $population_{size}$= *30* and the number of generations was $generation_{number}$ = 100. Regarding genetic operators parameters, crossover probability per chromosome was p_c= 0.8 and mutation probabil-

ity per gene was p_m= 0.015. These values were selected because of their better performance when compared with other combinations that were evaluated as well.

4.2 Post-nonlinear mixture.

For the PNL case, we used two uniform random signals as the sources (10.000 samples). Subsequently, we applied the following mixing matrix and nonlinearities in order to obtain the mixtures:

$$A = \begin{bmatrix} 1 & 0.5 \\ -0.5 & 1 \end{bmatrix}, f = \left[f_1(x) = \tanh(x),\ f_2(x) = \tanh(\frac{x}{2}) \right]. \quad \textbf{(9)}$$

Fig. 3 shows the joint distributions of the sources, post-nonlinear mixtures and estimations. The scatter plot of the mixtures (x) is not a result of only a linear deformation, but also a nonlinear transform. Estimations (y) recover in great part the original squared shape of the sources. The results of the error indexes were:

Crosstalk(y_1, s_1) (dB) = -23.90 dB, NormSQRMS(y_1, s_1) = 0.6385×10^{-3}
Crosstalk(y_2, s_2) (dB) = -23.13 dB, NormSQRMS(y_2, s_2) = 0.6971×10^{-3}

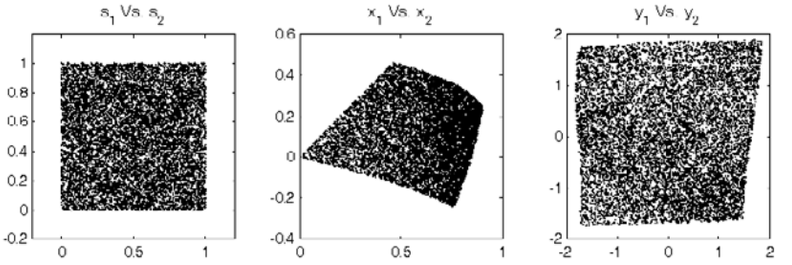


Fig. 3 Scatter plots (joint distributions) of sources, pnl-mixtures and estimations.

The parameters of the genetic algorithm in this experiment were the same as in Sec. 4.1., except for the population size that was increased to 40 ($population_{size}$= *40*).

5 Conclusion

This article presents a satisfactory and simple application of genetic algorithms to the complex problem of the blind separation of sources in linear and post-nonlinear mixtures through approximation of mutual information. It is widely believed that the specific potential of genetic or evolutionary algorithms originates from their parallel search by means of entire populations. In particular, the ability of escaping from local optima is an ability very unlikely to be observed in steepest-descent methods. Although to date, and to the best of the authors' knowledge, there is not much attention in the literature of this synergy between GAs and BSS in linear and nonlinear mix-

tures, the article shows how GAs provide a tool that is perfectly valid as an approach to this problem.

Acknowledgement

This work has been partially supported by the CICYT Spanish Projects TIC2001-2845 and DPI2001-3219.

References

1. S-I. Amari, A.Cichocki, H.Yang, A New Learning Algorithm for Blind Signal Separation, In Advances in Neural Information Processing Systems, vol.8, (1996).
2. A. Bell & T.J. Sejnowski: An Information-Maximization Approach to Blind Separation and Blind Deconvolution. Neural Computation 1129-59 (1995).
3. J.F. Cardoso, A. Souloumiac, "Blind beamforming for non Gaussian signals", IEEE Proceedings-F, vol. 140, no. 6, pp. 362-370, December 1993
4. P. Comon, Independent component analysis, a new concept?, Signal Processing, vol. 36, no. 3, pp. 287--314, 1994.
5. D.E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning",AddisonWesley, Reading, MA, (1989).
6. Hyvärinen, J. Karhunen y E.Oja, Independent Component Analysis. J. Wiley-Interscience. (2001).
7. A. Hyvärinen and P. Pajunen. Nonlinear Independent Component Analysis: Existence and Uniqueness results. Neural Networks 12(3): pp. 429-439, 1999.
8. C. Jutten and J.F. Cardoso. Source separation: really blind? In *Proc. NOLTA*, pages 79-84, 1995.
9. T-W.Lee, Independent Component Analysis: Theory and Applications, Kluwer Academic Publishers, (1998)
10. Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, New York USA, Third Edition, (1999).
11. C.G.Puntonet, A.Prieto, Neural net approach for blind separation of sources based on geometric properties, Neurocomputing, 18 (3), (1998), pp.141-164.
12. F. Rojas, I. Rojas, R.M. Clemente, C.G. Puntonet. "Nonlinear Blind Source Separation using Genetic Algorithms" Proceedings of the 3rd International Conference On Independent Component Analysis and Signal Separation (ICA2001). pp. 400-405, December 9-13, 2001, San Diego, CA, (USA)
13. I. Rojas, C.G. Puntonet, A. Cañas, B. Pino, J. Fernández, F. Rojas, "Genetic Algorithms for the Blind Separation of Sources" Proceedings of the 1st Artificial Intelligence and Applications (AIA), pp. 131-135, September 4-7, 2001, Marbella, (Spain).
14. A.Taleb, C.Jutten, "Source Separation in Post-Nonlinear Mixtures", IEEE Transactions on Signal Processing, vol.47 no.10, pp.2807-2820 (1999).
15. Y.Tan, J.Wang, Nonlinear blind source separation using higher order statistics and a genetic algorithm Ying Tan; Jun Wang; Evolutionary Computation, IEEE Transactions on, Volume: 5 Issue: 6 , pp. 600 -612 (2001)

B d ep f e d e f e e g e e e

S ss d ll

n r a Sa a r a ora o r co q ro og In r n a on
a R ro Nar onn 0 o o x ranc
ei i i t y e i e i t

A st t In a r w ro o an a roac or ara ng n ar
q a ra c x r o n n n r a o rc o a on
ara r c n ca on o a r c rr n ara ng r c r an
o an a a a gor w c g r or r a c o
o o r c r oca a o r c rr n
r c r an ow x r n a a w n a a ara
ng o n cc n ara ng o rc

n uc i n

l d s s p t SS s w d l st d d t t s
d ppl d s ss ll s d s s pl w
d s tl w l s s d t t s s
l t s t s w ll w t t t l s
t d p d p t s s s t s ffi t s p t l t s
s t l d t s w t l SS p
l ll p s d t l d d t d t s s t st t
st t s d s p t d ls

t s p p w st d s pl l d t t d l w
s d d s t s pl st l s l p l l d l
d l p d d s w l z d t pl t d p l l
d ls w p s t l s s pp t t l l
d ls

s n

S pp s d 2 tw d p d t d l s t ll w
l st t s t d l

2 2 2
2 2 22 2 2 2

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 241-248, 2003.

w w ld l t st t d 2 p t p t t d s l t
d p ss l dd t st t s pl t l t s d t d
2 22 2 t s w tt s

2 2

2 2 2 2 2

w 2 22 d 2 2 p s t t l t t s
t s s t t d 22 d 2 2 22 p
s t t d t t t s t s s s s pl
t t t s t s p t st t t ll w s t s t t t
2 t d l s d d t t w d l st d d l d l

l t d l t t dd t l t s 2 d
2 s st d d w t s sp t t s
w s s t t t s d l sp l l ss s s l pl
s s s st t ll ws t t s d 2 4 2 d 2 2
s t dd t l s s s s p t t ds s d p p t
l s s d t w ll t ss ts t s s
2 4 z Us t l st l t s t s s s t
s p t d s pl t d l z t p d
l t t s d l SS t t w w w s pp s t t
t s s t l s ls d l tw t s l l

nv i i i y n s p in s uc u

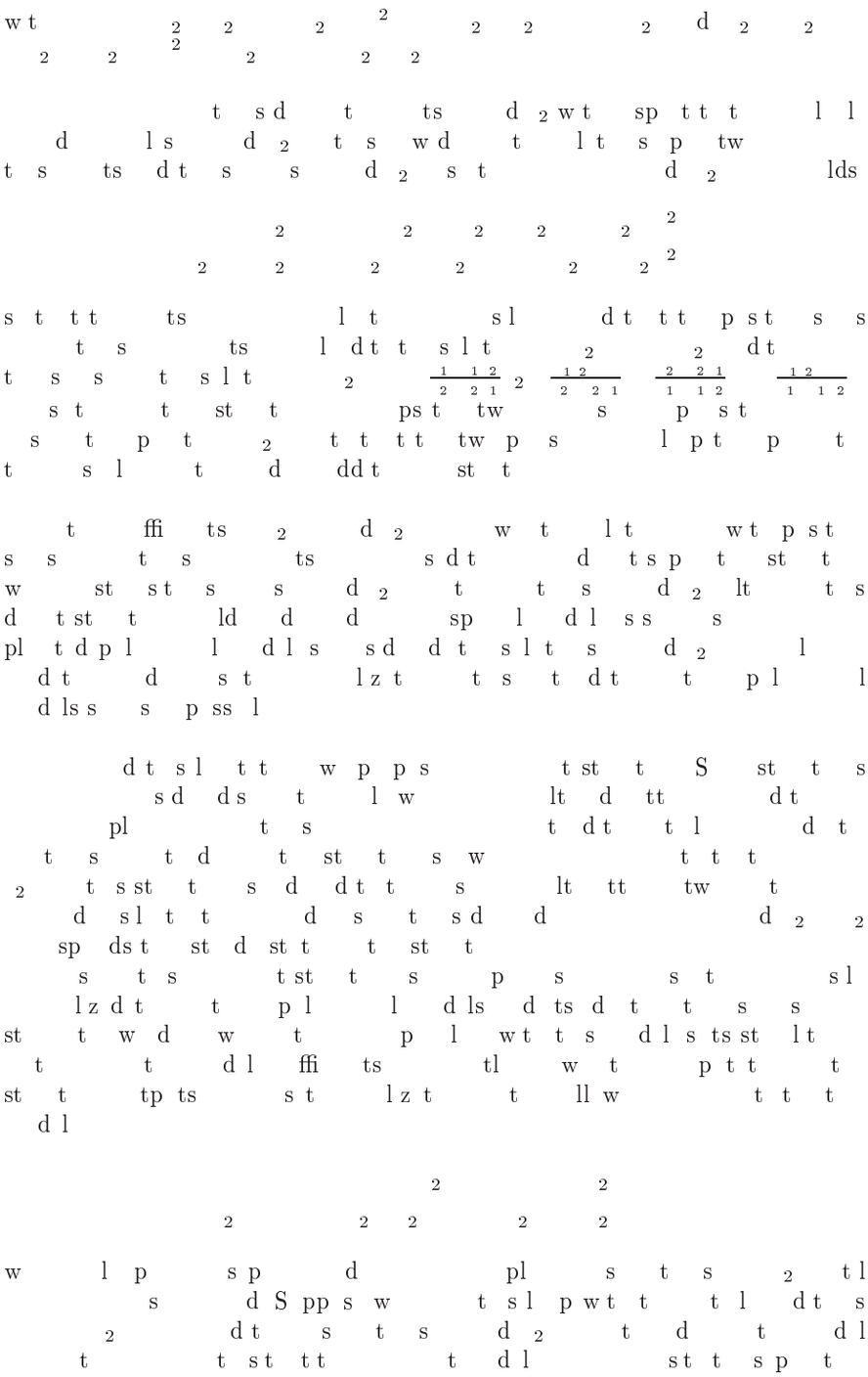
st st p st t s stss l z t t l t t
s d d l ss t s d t w t t s l s d
2 d d t s d t s d 2 w ffi ts
2 d 2 lt pl t st t 2 d t s d
t t dd t s lts lds

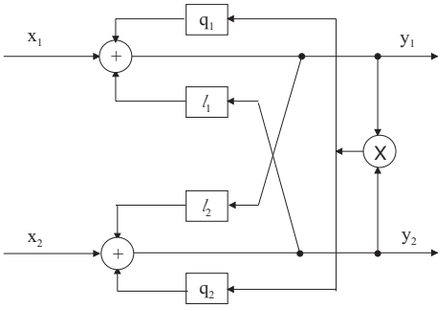
2 2 2 2 2 2
s st t t 2 t s d t l ds t ll w s d d
t

2 2 2 2 2 2 2
S l l t sp d t w t sp t t 2 s

2 2 2 2 2 2 2 2 2
t s l t t t s t s z tw p ss l l
l s l t s S l t t s w t

2 2 2 2 2
2 2
2 2 2 2 2 2
2
2





F R c rr n ara ng r c r

p t 2 t t t d l d p ds t
ffi t l s d t l s s s
d ffi lt t l s t ll w w st d l t ss d
s ffi t d t s l l st l t w s d d s ss
d t s l l st l t

c s i i y f cu n

st d t l l st l t t d l t t d p t 2 w
l z t d l t t s p t t s w t t t **y**
y w **y** 2 d s t tw d s l t d d
z t s t t t d p t 2 lds

y **y**

w s t t d d s $\frac{n}{n}$ d l t d t
d t s l t t t w t d l
z **y** t w t s

z **z**

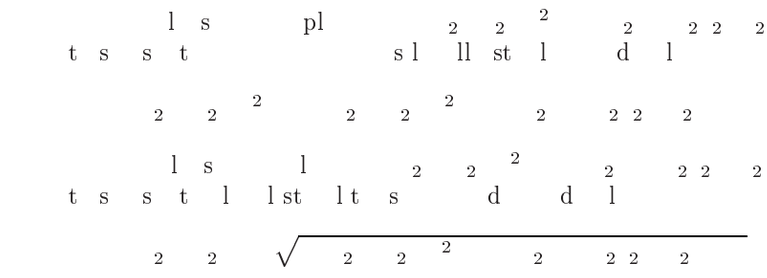
y l ll s t w d l **z** s t
z d t d l t t s t t
l $_n \infty ^n$ ss d s ffi t d t s t t t s l t l s
t tw l s s ll t s d t
t t s

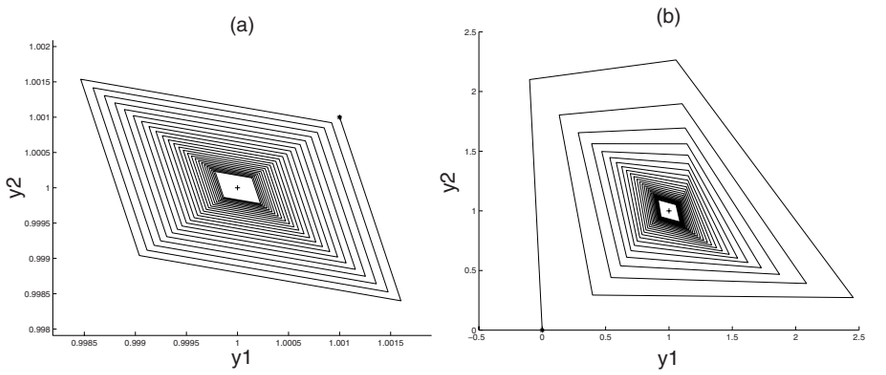
2
2 2 2 2

d ts l s

– 2 2 2 2 2 2 2 2 2 2

w d ff t s s s d d





F 2 on rg nc ra c or a w n a con on n n g or oo o ac a o rc w ro con on

u c s p i n i

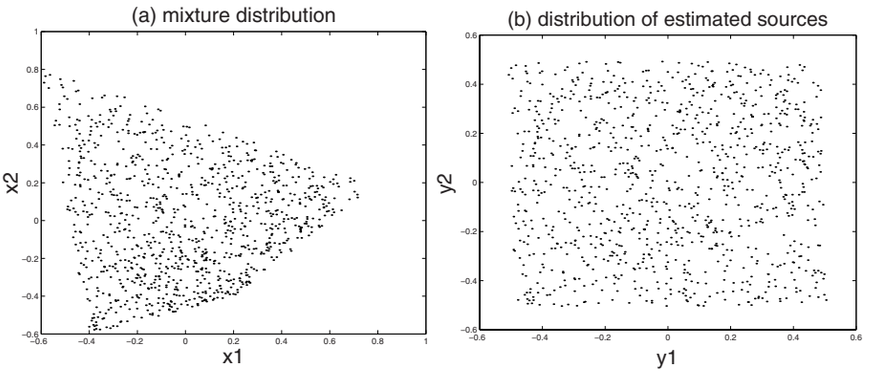
d pt l t t w p p s t t s d s ls sp d t l
w lt d tt l t s d t s s t l
t t tw p t s d 2 d t d s tw l d l
t t s *f* 2 d *f* 2 w t
d t p t s w d l d l t t s pt l
t t s *f* d t s t s s d t t
l l d pp t t d ppl d d
ll t l t t st t t s t t
w s t s l t p p w p t t s lts t d sp l
t s t s w d pt t l p t s d 2 t
sp t l 2 d 2 d t d t p t s
d 2 t t sp t l 2 2 d 2 w d 2
p s t t t d s s t tp ts t s p p s w s d t
ll w t l t w **x** d $\mathbf{x}_2$ t t s t t
s t s pl s **y** d $\mathbf{y}_2$ t t s st t d s s s p s t
d pt t d d t p d ts d p w s p t d p t w s

y $\mathbf{y}_2$
l 2 *d* 2 *o* *ll* *do* *lu*
R
R
y e **x** $\mathbf{y}_2$ **y** $\mathbf{y}_2$
y e 2 $\mathbf{x}_2$ 2**y** 2**y** $\mathbf{y}_2$
y **y e**
$\mathbf{y}_2$ **y e** 2
U l o *of* *u* *u u*
y $\mathbf{y}_2$ $\mathbf{y}_2$
2 2 $\mathbf{y}_2$ **y** **y**
$\mathbf{y}^2$ $\mathbf{y}_2$ $\mathbf{y}_2$
2 2 $\mathbf{y}_2^2$ **y** **y**
U l o *of* *o*

i u i n su s

s l t s d s s pl s tw d s s l
d st t d ll t p ts t p t s
s t s t t t d l d st t
t t s d 2 s pl p t w t 2
d 2 s s w l l t
t t s s s ws t d st t t st t d
s s d 2 ppl t p p s d l t w d t s l l t t
t d p d t p ts t d

s p t w s p t d t s sp d t d ff t s d
l s d s t p t t S l t s



F a r on o wo x r r on o wo a o rc

t p t S s d s p t d

$$S \quad R \quad \text{l} \quad \frac{2}{2}$$

d t t t l S R s d d s t t tw S R d t st d d t S R t p ts w s d d d sp t l s lt s t d p t s p t l t

p t t p t w t t ffi t l s l st l d t t s s lts w t p l s t t t d pt t p d s t s t ffi ts t w t t tw s t st l st d pt z t pp s s ss t d t s s t t s p ts w t t s st t st s ls ss t t s l t s lts

C nc usi n

s p p p s t d st pp s p t l d t t s d p d t l s s s t s p t tw d t s tw s t t t l z d t pl t d p l l d ls U t t l t st l t l s s s tw s s d lt t s pl s st d d t p p p t d l l st l t l s s ll ws s t d t l ll ss d s ffi t d t s w p s s st ts t t ffi ts

st p t l s lts p s d s w t t t tw s ds s p t t s s w t d p ts st l t s t d tl w t pl t t w p t d t s p p

S t s p s l s s t st l t t
t tw d t t pt l l d l t t s d
st d p t d pt t l t s l s s s
p s st t s s l l pp s 6
p ts w t d ff t s st t st s d d ff t t ffi ts d
l z t t t d t pl t d p l l d ls d
s s

f nc s

ar n n J ar n n an a *n n n n a y* J
00
c oc an S I ar *a n S gna an ag c ng a n ng*
g an ca n J 00
ar n n an a n n Non n ar n n n co on n ana x nc
an n q n r *a* o no
a an J n So rc ara on n o non n ar x r *an*
n S gna c ng o no 0 0 0
a n ar an non n ar I a on a n or a on In *c*
a Sy S gna c ng n ca n an n Sy
S S a o ana a c o r 000
J r on an o n n n n a o c a o non n ar n an an o
I o In *c an S gna c n S*
o 0 o o ranc S r 00
S o n an J n n ara o non n ar x r o o
ra corr a o rc *S gna c ng* o 0 no
r ar 00
J n a a a S o n r a wa or ara ng non n ar
x r o *S gna c ng*
ro an n r n n ca on o a n ar q a ra c o ng
g r or r a c In *c SS* o 0
0 ra o c ran an a n n ca on o a n ar
q a ra c x r o n n n co on n a on o n agona a on ro
c r In *c SS* an a S a
J n an J ra n ara on o o rc ar I n a a a gor
a on n ro c arc c r *S gna c ng* 0
Ng n an J n n o rc ara on or con o x r *S gna*
c ng o no 0
N ar an an D S a a ara on o con o x gna
w a r c r r c r ar I S a ana an o a on o a o c
a o r *S gna c ng* o n
N ar an an D S a a ara on o con o x g
na w a r c r r c r ar II or ca x n on an a ca on o
n c an r a gna *S gna c ng* o n 0
ra Gar n ar go a an J a a a *a c yna c n*
n na n n n b a or Sc n c S r on Non n ar Sc nc
o 0
D na o con rg nc ro r o nor a o rc a
ra on n ra n wor *an ac n n S gna c ng* o n
a

Advances in Neyman-Pearson Neural Detectors Design*

Diego Andina[1], Santiago Torres-Alegre[1], Antonio Vega-Corona[1], and Antonio Álvarez-Vellisco[2]

[1] Universidad Politécnica de Madrid
Departamento de Señales, Sistemas y Radiocomunicaciones, E.T.S.I. Telecomunicación
andina@gc.ssr.upm.es
[2] Universidad Politécnica de Madrid
Departamento de Ingeniería de Circuitos y Sistemas, E.U.I.T. Telecomunicación
aalvarez@ics.upm.es

Abstract. This chapter is dedicated to scope of the application of Importance Sampling Techniques to the design phase of Neyman-Pearson Neural Detectors. This phase usually requires the application of Monte-Carlo trials in order to estimate some performance parameters. The classical Monte-Carlo method is suitable to estimate high event probabilities but not suitable to estimate very low event probabilities (say, 10^{-4} or less). For estimations of very low false-alarm probabilities (or error probabilities), a modified Monte-Carlo technique, so-called Importance Sampling (IS) technique, is then considered.

1 Introduction

Neyman-Pearson Neural Detectors (NP-NDs) are the neural alternative to binary detectors optimized in the Neyman-Pearson sense [1]. These detectors present a configurable low probability of classifying binary symbol *1* when symbol *0* is the correct decision. This kind of error, referred in the scientific literature as *false-positive* or *false alarm probability* has a high cost in many real applications as medical Computer Aided Diagnosis [2] or Radar and Sonar Target detection [3], and the possibility of controlling its maximum value is crucial. More specifically, we are dealing with Monte-Carlo simulations needed to estimate meaningful parameters by repetition of trials in the computer. The parameters to be estimated are: detection and false-alarm probabilities (or error probability). The IS technique presented has been previously introduced in other publications about general MLP [3] and Neural detectors, and now it is detailed for the NP-NDs, where the technique is successfully arising in several design aspects.

* This research has been supported by the National Spanish Research Institution "Comisión Interministerial de Ciencia y Tecnología-CICYT" as part of the project TIC2002-03519

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 249–256, 2003.

This paper is organized basically into two sections. Section one describes the NP-ND parameters relevant to the application of the Importance Sampling (IS) technique and section two describes the IS technique itself.

1.1 Neyman-Pearson Neural Network Detector

The advanced IS technique that is going to be described is useful to estimate low probabilities in the Neural detector. The NP-ND inputs are samples of the complex envelope in a sequence of M complex samples. The two detection hypotheses are described as follows

$$\begin{aligned} H_0 &: \tilde{x}(k) = \tilde{n}(k) \\ H_1 &: \tilde{x}(k) = S \cdot e^{j\varphi(k)} + \tilde{n}(k) \end{aligned} \tag{1}$$

where k varies from 1 to M, $\tilde{x}(k) = x_c(k) + jx_s(k)$ is the input complex envelope, S is the signal amplitude (constant), $\varphi(k)$ is the phase, and $\tilde{n}(k)$ represents an uncorrelated zero-mean Gaussian complex sequence of noise with variance σ^2 in each component. Each complex input is separated in its real x_c and imaginary x_s parts, yielding two real inputs to the NN; so, the number of input nodes must be $2M$.

The Signal-to-Noise Ratio (SNR) is defined in dB as

$$SNR = 20 \log \left(\frac{S}{\sigma} \right) \tag{2}$$

Because $\tilde{n}(k)$ is uncorrelated (white) and Gaussian, the input pdf under the null hypothesis H_0 (target absent) is given by

$$f_{\bar{X}}(\bar{x}|H_0) = (2\pi)^{-M} \exp \left(-1/2 \sum_{i=1}^{M} (x_{ci}^2 + x_{si}^2) \right) \tag{3}$$

where $\bar{x} = (x_{c1}, x_{s1}, x_{c2}, x_{s2}, \ldots, x_{cM}, x_{sM})$ and M is the number of input complex samples (or number of pulses per antenna beamwidth). Also, unit noise variance ($\sigma^2 = 1$) was supposed without loss of generality.

The input pdf under the alternative hypothesis H_1 (target present), defined in (1) and supposed Marcum's model for the target [1], is given by

$$f_{\bar{X}}(\bar{x}|H_1) = \int_0^{2\pi} f_{\bar{X}}(\bar{x}|H_1, \varphi) f(\varphi) d\varphi \tag{4}$$

where

$$f_{\bar{X}}(\bar{x}|H_1, \varphi) = (2\pi)^{-M} \exp \left(-\frac{1}{2} \sum_{i=1}^{M} \left\{ (x_{ci} - S\cos\varphi)^2 + (x_{si} - S\sin\varphi)^2 \right\} \right) \tag{5}$$

and $f(\varphi)$ is the uniform distribution in $[0, 2\pi]$, i.e. $f(\varphi) = 1/2\pi$, if $0 < \varphi < 2\pi$, and $f(\varphi) = 0$, otherwise.

2 Importance Sampling (IS) Technique

Let us consider a nonlinear system with a system function $g(\cdot)$, an input vector $\bar{x} \in \mathbb{R}^n$, where $\mathbb{R}^n$ is the n-dimensional space, i.e. $\bar{x} = (x_1, x_2, \cdots, x_n)$ with $x_i \in \mathbb{R}$, $i = 1, 2, \cdots, n$, and a scalar output $y \in \mathbb{R}$; the system maps $\bar{x}$ into y by means of $g(\cdot)$. Now, if you consider the random variable $\bar{X} = (X_1, X_2, \cdots, X_n)$ with probability density function (pdf) $f_{\bar{X}}(\bar{x}) = (x_1, x_2, ..., x_n)$, the system $g(\cdot)$ induces a random variable $Y = g(\bar{X})$ with probability density function $f_Y(y)$.

Define the input pdf under a hypothesis H_i by $f_{\bar{X}}(\bar{x}|H_i)$, and the output pdf by $f_Y(y|H_i)$, $i = 0, 1$. Now, suppose that the output of the system is thresholded in order to perform a binary detection; then $z = u(y - T_0)$, where T_0 is the threshold and $u(\cdot)$ is the unit-step function, i.e. $u(t) = 1$ if $t \geq 0$, $u(t) = 0$ if $t < 0$. The decision rule is: if $z = 1$, it is supposed that hypothesis H_1 is true (target present); if $z = 0$, the decision is H_0 (target absent).

The detection probability $P_d \triangleq P_1$ and false-alarm probability $P_{fa} \triangleq P_0$ are defined [4] by

$$P_i = Pr\{Z = 1|H_i\} = \int_{T_0}^{\infty} f_Y(y|H_i)dy, \ i = 0, 1 \tag{6}$$

This definitions of P_d and P_{fa} are important in the context of radar or sonar detection. In the context of communications, the error probability P_e is the fundamental parameter, and if the two hypotheses (symbols) are equally likely i.e. $Pr\{H_1\} = Pr\{H_0\} = 1/2$, then $P_e = (1 - P_1)/2 + P_0/2$. Very often formulas (6) are difficult to compute because output pdf $f_Y(\cdot)$ under each hypothesis does not have a well known analytical expression. On the other hand, formulas 6 can be expressed by means of input pdf $f_{\bar{X}}(\cdot)$, as follows:

$$P_i = E\{u(g(\bar{X}) - T_0)|H_i\} = \int_{\mathbb{R}^n} u(g(\bar{x}) - T_0) f_{\bar{X}}(\bar{x}|H_i)d\bar{x}, \ i = 0, 1 \tag{7}$$

where $E\{u(g(\bar{X})) - T_0)|H_i\} = E\{Z|H_i\}$ means mathematical expectation of the random variable Z conditioned by hypothesis H_i. Remember that the false-alarm probability $P_{fa} = P_0$, and the detection probability $P_d = P_1$.

Formulas (7) are equivalent to formulas (6), however in (7) the pdf's correspond to the input $\bar{X} = (X_1, X_2, \cdots, X_n)$, and it is supposed that these pdf's are well known. On the other hand, $u(g(\bar{x}) - T_0) = 1$ defines the decision region $g(\bar{x}) \geq T_0$ in the $\mathbb{R}^n$-space that corresponds to the acceptance of hypothesis H_1 or the rejection of H_0; but, in general, the boundary surface $g(\bar{x}) = T_0$ is too complex, where it is suppose the threshold T_0 is known. Consequently, formulas 7 are not computable in most cases, so that we have to use statistical techniques to estimate $P_i, (i = 0, 1)$, being an approach as follows.

It is well known that a good P_i-estimator is given by [5, 6, 7, 8]:

$$\hat{P}_i = \frac{1}{N} \sum_{k=1}^{N} u(g(\bar{x}_k^{(i)}) - T_0), i = 0, 1 \tag{8}$$

where $\bar{x}_k^{(i)} = (x_{1k}^{(i)}, x_{2k}^{(i)}, \cdots, x_{nk}^{(i)}), k = 1, 2, \cdots, N$, are independent sample vectors of the input random vector $\bar{X} = (X_1, X_2, \cdots, X_n)$ having $f_{\bar{X}}(\bar{x}|Hi)$ as pdf, $i = 0, 1$.

The relative error $\varepsilon_{\hat{P}_i}$ of the estimator $\hat{P}_i$ is defined by [5]

$$\varepsilon_{\hat{P}_i} = \frac{\sigma_{\hat{P}_i}}{\mu_{\hat{P}_i}} = \sqrt{\frac{1 - P_i}{N \cdot P_i}},\ i = 0, 1 \tag{9}$$

where $\sigma_{\hat{P}_i}$ is the standard deviation of the estimator, and $\mu_{\hat{P}_i}$ is the mean value. Formula (9) is fundamental in Monte-Carlo trials, because we can estimate the number of samples required for the simulation. For example, if $\varepsilon_{\hat{P}_i} = 0.1$ (i.e. 10%) and $P_i = 0.5$, then $N = 100$; if $\varepsilon_{\hat{P}_i} = 0.01$ (1%) and $P_i = 0.5$, then $N = 10^4$ samples. Similar results for the N-values can be obtained from confidence interval theory.

Finally, if $P_{fa} \ll 1$, from 9 we have $N \simeq (\varepsilon_{\hat{P}_{fa}})^{-2}/P_{fa}$, i.e. the number of samples required in the simulation increases as the false-alarm probability decreases, e.g. if $\varepsilon_{\hat{P}_{fa}} = 10\%$, $P_{fa} = 10^{-6}$, then $N \simeq 10^8$ samples. Classical Monte-Carlo techniques fail in the estimation of very low false-alarm probabilities because of the large time required in the computer simulations. Consequently, a modified Monte-Carlo technique called "Importance Sampling" (IS) is considered for solving this computational problem.

Consider a new probability density function (pdf) $f^*_{\overline{X}}(\overline{x})$,such that for any $\overline{x}_0 \in \{\overline{x} : g(\overline{x}) \geq T_0\}$ if $f_{\overline{X}}(\overline{x}_0|H_0)\ /= 0$, then$f^*_{\overline{X}}(\overline{x}_0)\ /= 0$. Now, from 7) we can write

$$P_{fa} = \int_{\mathbb{R}^n} u(g(\bar{x}) - T_0) \frac{f_{\overline{X}}(\overline{x}_0|H_0)}{f^*_{\overline{X}}(\overline{x})} f^*_{\overline{X}}(\overline{x}) d\bar{x} = E^*\{w(\bar{X}) \cdot u(g(\bar{X}) - T_0)\} \tag{10}$$

where $E^*\{\cdot\}$ means expectation with respect to $f^*_{\overline{X}}(\overline{x})$, and $w(\bar{x})$ is the weighting function, defined as follows

$$w(\bar{x}) = \frac{f_{\bar{X}}(\bar{x}|H_0)}{f^*_{\bar{X}}(\bar{x})} \tag{11}$$

The sample mean of $w(\bar{X}) \cdot u(g(\bar{X}) - T_0)$ under pdf $f^*_{\bar{X}}(\bar{x})$ is an estimator of P_{fa}, i.e.

$$P^*_{fa} = \frac{1}{N} \sum_{k=1}^{N} w(\bar{x}^*_k) \cdot u(g(\bar{x}^*_k) - T_0) \tag{12}$$

where $\bar{x}^*_k$, $k = 1, 2, \cdots, N$, are independent sample vectors from the input random vector $\hat{X}$ whose pdf is $f^*_{\bar{X}}(\bar{x})$ (also referred as IS pdf). If $w(\hat{x}) = 1$, i.e. $f^*_{\bar{X}}(\bar{x}) = f_{\bar{X}}(\bar{x}|H_0)$, then (12) is identical to (8) under H_0, which is the classical Monte-Carlo method. Also, under some mild conditions, the distribution of P^*_{fa} is asymptotically normal (Gaussian).

Now, we can calculate the mean $\mu_{P^*_{fa}}$ and the variance $\sigma^2_{P^*_{fa}}$ of the estimator P^*_{fa}. From (12) and (11), we have [6, 7, 8]

$$\mu_{P^*_{fa}} = E^*\{P^*_{fa}\} = P_{fa} \tag{13}$$

$$\sigma^2_{P^*_{fa}} = E^*\{(P^*_{fa} - P_{fa})^2\} = \frac{1}{N} \cdot \left(E^*\{w^2(\bar{X}) \cdot u(g(\bar{X}) - T_0)\} - P^2_{fa}\right) \tag{14}$$

Because the variance is not a negative number, from (14) we have

$$E^*\{w^2(\bar{X}) \cdot u(g(\bar{X}) - T_0)\} \geq P^2_{fa} \tag{15}$$

The equality case in (15) is satisfied if

$$f^*_{\bar{X}}(\bar{x}) = \frac{u(g(\bar{x}) - T_0)}{P_{fa}} \cdot f_{\bar{X}}(\bar{x}|H_0) \tag{16}$$

that can be proved by taking (16) into (15); then, the estimator variance in (14) is zero. Expression (16) is the unconstrained optimal solution for $f^*_{\bar{X}}(\bar{x})$, and it will be the reference for other suboptimal solutions.

Finally, the relative error $\varepsilon_{P^*_{fa}}$ of the estimation of P_{fa} is defined by

$$\varepsilon_{P^*_{fa}} = \frac{\sigma_{P^*_{fa}}}{\mu_{P^*_{fa}}} = \sqrt{\frac{1}{N} \cdot \left(\frac{E^*\{w^2(\bar{X}) \cdot u(g(\bar{X}) - T_0)\}}{P^2_{fa}} - 1\right)} \tag{17}$$

If $w(\bar{x}) = 1$, then (17) is identical to (9) under H_0.

The fact that (16) is satisfied, i.e. $f^*_{\bar{X}}(\bar{x}) = f_{\bar{X}}(\bar{x}|H_0)/P_{fa}$ for $g(\bar{x}) \geq T_0$, and $f^*_{\bar{X}}(\bar{x}) = 0$ for $g(\bar{x}) < T_0$, means that the variance of P^*_{fa} given in (14) is zero, i.e. the pdf of the estimator P^*_{fa} is the Dirac delta function $\delta(P^*_{fa} - P_{fa})$; also, from (17), $\varepsilon_{P^*_{fa}} = 0$ and the number of samples required in the estimator P^*_{fa} is only one ($N = 1$).

On the other hand, the unconstrained optimal solution for $f^*_{\bar{X}}(\bar{x})$ given by (16) is not realistic, because P_{fa} is not known "a priori" (it has to be estimated by (12); furthermore, if you set a fixed P_{fa}, then the threshold T_0 is unknown. Suboptimal solutions for $f^*_{\bar{X}}(\bar{x})$ which partially minimize (14) are given in the literature, by scaling input signals [5], by shifting them [6, 7], or by other strategies [8, 9, 10].

2.1 Suboptimal IS Density Function

From (13) and (14), $P^*_{fa} \to P_{fa}$ as $N \to \infty$ (i.e. P^*_{fa} is a consistent estimator of P_{fa}). However, the unbiasedness of P^*_{fa} shown in (13) requires the condition: $f^*_{\bar{X}}(\bar{x}) \not\equiv 0$ whenever $f^*_{\bar{X}}(\bar{x}|H_0) \not\equiv 0$ in the region $g(\bar{x}) \geq T_0$; if this condition is not satisfied, the estimator is biased (i.e. $E^*\{P^*_{fa}\} \not= P_{fa}$) and, more exactly, P^*_{fa} is an underestimator of P_{fa} (i.e. $E^*\{P^*_{fa}\} < P_{fa}$); consequently, $P^*_{fa} \to E^*\{P^*_{fa}\} < P_{fa}$ as $N \to \infty$. This important fact is not well clarified in the literature, and it is fundamental in applications.

In the case of optimum detection, the system function $g(\bar{x})$ of Figure 1, is the likelihood ratio statistic [4], i.e.

$$g(\bar{x}) = \frac{f_{\bar{X}}(\bar{x}|H_1)}{f_{\bar{X}}(\bar{x}|H_0)} \tag{18}$$

or any monotonic increasing function of (18); moreover, some researchers [9, 10] propose $f_{\bar{X}}(\bar{x}|H_1)$ as IS density function. As in the decision region of H_1 ($g(\bar{x}) \geq T_0$) it is fulfilled that $f_{\bar{X}}(\bar{x}|H_1) > T_0 \cdot f_{\bar{X}}(\bar{x}|H_0)$ whenever $T_0 > 0$, we conclude that the unbiasedness condition for P^*_{fa} (i.e. $E^*\{P^*_{fa}\} = P_{fa}$) is satisfied if $f^*_{\bar{X}}(\bar{x}) = f_{\bar{X}}(\bar{x}|H_1)$.

Furthermore, for a practical detector (not necessarily optimum, e.g. neural detector) with realistic P_{fa} and P_d values ($P_{fa} \ll P_d$) it is verified that $f_{\bar{X}}(\bar{x}|H_0) \ll f_{\bar{X}}(\bar{x}|H_1)$ for almost all $\bar{x}$-values in the region $g(\bar{x}) \geq T_0$. Consequently, if we choose$f^*_{\bar{X}}(\bar{x}) = f_{\bar{X}}(\bar{x}|H_1)$, then from (11) we have $w(\bar{x}) \ll 1$ in $g(\bar{x}) \geq T_0$, and this IS technique is much more efficient than the classical Monte-Carlo one ($w(\bar{x}) = 1$).

Frequently H_1 is a composite hypothesis [4] (i.e. $f_{\bar{X}}(\bar{x}|H_1;\theta)$ is a parametric family with parameter θ), so we choose a $f_{\bar{X}}(\bar{x}|H_1;\theta^*)$ in H_1 as IS density function in such a way that (14) or (17) is minimized. Now, the question is: How can we find this optimal $f_{\bar{X}}(\bar{x}|H_1;\theta^*)$? Theoretically, in complex detectors (e.g. neural detectors) it is very difficult, if not impossible; experimentally, it is very easy by considering an estimator of (17), as follows.

An error estimator $\hat{\varepsilon}_{P^*_{fa}}$ of (17) can be expressed as

$$\hat{\varepsilon}_{P^*_{fa}} = \sqrt{\frac{1}{N} \cdot (\Delta^2 - 1)} \tag{19}$$

where the statistic Δ^2 is an estimator of $E^*\{w^2(\bar{X}) \cdot u(g(\bar{X}) - T_0)\}/P^2_{fa}$, and it is given by

$$\Delta^2 = \begin{cases} \frac{\frac{1}{N}\sum_{k=1}^{N} w^2(\bar{x}^*_k) \cdot u(g(\bar{x}^*_k) - T_0)}{(P^*_{fa})^2} & \text{if } P^*_{fa} \neq 0 \\ N & \text{if } P^*_{fa} = 0 \end{cases} \tag{20}$$

with P^*_{fa} given in (12). Also, it can be proved from the Schwartz inequality that $1 \leq \Delta^2 \leq N$.

The statistic $\hat{\varepsilon}_{P^*_{fa}}$ of (19) is the key for the optimization of our Importance Sampling Technique, and controls the error of the P_{fa}-estimator. There are other equivalent statistics [11] to optimize IS parameters, but we suggest $\hat{\varepsilon}_{P^*_{fa}}$ as one very easy to compute and directly related to the number of samples N required for a specific estimation error.

The statistic Δ^2 of (20) is close to the unit if the probability density function $f^*_{\bar{X}}(\bar{x})$ for the Importance Sampling is close to the unconstrained optimum given in (16). In order to find the best $f^*_{\bar{X}}(\bar{x})$ in the $f^*_{\bar{X}}(\bar{x}|H_1;\theta)$ family, we compute Δ^2 according to (20) for a simulation run (an algorithm execution on the computer) with a specific θ-value. After some simulation runs (computations of estimator formulas) for different θ-values, the minimum $(\Delta^2)_{min}$ can be found for the IS density $f^*_{\bar{X}}(\bar{x}|H_1;\theta^*)$, and from (19), we compute the minimum error $(\hat{\varepsilon}_{P^*_{fa}})_{min}$; e.g. if $5 < (\Delta^2)_{min} < 50$ for $N = 10^4$ samples, then $2\% < (\hat{\varepsilon}_{P^*_{fa}})_{min} < 7\%$. Note that $f^*_{\bar{X}}(\bar{x}|H_1;\theta^*)$ is dependent on the value of P_{fa} to be estimated, i.e. for each P_{fa} there is a different $f^*_{\bar{X}}(\bar{x}|H_1;\theta^*)$ in

the $f^*_{\bar{X}}(\bar{x}|H_1;\theta)$ family; then, it is very important to optimize the algorithm required for implementing the Importance Sampling.

From (5) and (4) after some manipulations, we have

$$f_{\bar{X}}(\bar{x}|H_1) = \\ = (2\pi)^{-M} \cdot \exp\left(-1/2 \sum_{i=1}^{M} (x_{ci}^2 + x_{si}^2 + S^2)\right) \cdot \\ \cdot I_0\left(S \cdot \left\{ (\sum_{i=1}^{M} x_{ci})^2 + (\sum_{i=1}^{M} x_{si})^2 \right\}^{1/2}\right)$$

where $I_0(\cdot)$ is the modified Bessel function of first kind and order zero, S is the parameter of the pdf family, and (4) becomes (3) if $S = 0$. Also, S is the SNR defined in 2 because $\sigma = 1$.

The threshold T_0 has to be estimated for a fixed false-alarm probability P_{fa}, using Importance Sampling Technique for the cases of $P_{fa} < 10^{-3}$. Taking into account the alternative hypothesis H_1 defined in (1), the IS probability density function $f^*_{\bar{X}}(\bar{x}) = f^*_{\bar{X}}(\bar{x}|H_1)$ is given by (21). Now, taking (3) and (4) into (11) we obtain $w(\bar{x})$

$$w(\bar{x}) = \frac{\exp(-S^2 M/2)}{I_0\left(S \cdot \left\{ (\sum_{i=1}^{M} x_{ci})^2 + (\sum_{i=1}^{M} x_{si})^2 \right\}^{1/2}\right)} \tag{21}$$

where $\bar{x} = (x_{c1}, x_{s1}, x_{c2}, x_{s2}, \ldots, x_{cM}, x_{sM})$, M is the number of integrated pulses and S is the IS-parameter.

3 Conclusions

An efficient IS algorithm has been proposed to be applied in Neyman-Pearson Neural Detector simulations.

Note that IS techniques improve MC simulations for the following reasons: (a) IS techniques save time in the computer simulations. (b) IS techniques avoid the generation of random variables in the distribution tails, which cannot be properly simulated in the computers. (c) IS techniques provide an alternative to estimate parameters of one distribution by means of random variables of another distribution.

Finally, in order to accelerate the convergence of training, it is very interesting to apply Importance Sampling techniques in the training phase, taking into account the appropriate modifications on criteria and objective functions. This subject is now producing promising results in combination with Genetic Algorithms [12]. Its integration in MLP general training is also producing the expected results in the sense of training acceleration of Backpropagation algorithm, results that are on research and hopefully will be published in our next contribution.

References

[1] D. Andina, J. L. Sanz-González, Neyman-Pearson Neural Detectors, *Bio-Inspired Applications of Connectionism. Lecture Notes in Computer Science. Springer Verlag, Eds: J. Mira, A. Prieto*, Berlin-Heidelberg, 2085:554-557, 2001. 249, 250

[2] D. Andina, A. Vega-Corona, Detection of Microcalcifications in Mammograms by the Combination of a Neural Detector and Multiscale Feature Enhancement, *Bio-Inspired Applications of Connectionism. Lecture Notes in Computer Science. Springer Verlag, Eds: J. Mira, A. Prieto*, Berlin-Heidelberg, 2085:385-392, 2001. 249

[3] D. Andina, F. Ballesteros, *Recent Advances in Neural Networks*, Ed. International Institute of Informatics and Systemics, IIIS press, Illinois, 2000. 249

[4] H. L. Van Trees, *Detection, Estimation and Modulation Theory, Part I*, Eds. Wiley and Sons, New York, 1968. 251, 253, 254

[5] K. S. Shanmugam, P. Balaban, A modified Monte-Carlo simulation technique for the evaluation of error rate in digital communication systems *IEEE Trans. Commun.*, vol. COM-28, pp. 1916-1924, Nov. 1980. 251, 252, 253

[6] D. Lu, K. Yao, Improved importance sampling technique for efficient simulation of digital communication systems, *IEEE J. Select. Areas Commun.*, vol. SAC-6, pp. 67-75, Jan. 1988. 251, 252, 253

[7] R. J. Wolfe, M. C. Jeruchim, P. M. Hahn, On optimum and suboptimum biasing procedures for importance sampling in communication simulation *IEEE Trans. Commun.*, vol. COM-38, pp. 639-647, May 1990. 251, 252, 253

[8] J. C. Chen, D. Lu, J. S. Sadowsky, K. Yao, On importance sampling in digital communications. Part I: Fundamentals *IEEE J. Select. Areas Commun.*, vol. SAC-11, pp. 289-299, Apr. 1993. 251, 252, 253

[9] G. C. Orsak, A note on estimating false alarm rates via importance sampling, *IEEE Trans. Commun.*, vol. COM-41, pp. 1275-1277, Sept. 1993. 253, 254

[10] J. S.Stadler, S.Roy, Importance sampling for detection of scale problems, *IEEE Trans. Commun.*, Vol. COM-43, pp. 2862-2865, Dec. 1995. 253, 254

[11] M. Devetsikiotis, J. K. Townsend, An algorithmic approach to the optimization of importance sampling parameters in digital communication system simulation, *IEEE Trans. Commun.*, vol. COM-41, pp. 1464-1473, Oct. 1993. 254

[12] J. L. Sanz-González, D. Andina, J. Seijas, Importance Sampling and Mean-Square Error in Neural Detector Training, *Neural Processing Letters, Kluwer academic publishers*, 16:3:259-276. Dec. 2002. 255

A e pe ed egy ep e e e e f e e y d

d p l s s d

D ar a n o c ron ca S a
n r a a or ña
a ña /n 0 or ña S IN
000 ax 0 a a r ana c *

A st t In a r w ro o a n w ra g o ara con o x r o ora w gna a c a o ran or con o x r n ra n an an o x r ng cr o r r ran or S q n ac n an an o x r ara ng a n ra n wor w o co ffic n ar a a n ng an q ar rror w n o an a r gna r o o a n ng an n r a gor J D n w ra g o no ff r ro a / r a on n r nac a a ar n o r r q nc o a n a roac *y* n o rc ara on con o x r n con o on r co n ca on

n uc i n

l d S S p t SS s sts st t st ll d p d t s ls s s t s t st s ls s s w t t w t s s t s st s p l s w d l st d d t t t l tw s s SS l t s t p t d s s p s d l l s 6

st SS l t s d l p d ss t t s t s s w t d s s t s s U t t l t p t s s st t s t s l st p t l ppl t s w t s s t lt pl p t s d t s d t t d l t s t s s l t t s s l p l s l d t t d s t p ls sp s lt s t d t p t t l t t s s s l st t s t s t d t d SS SS pp s s t t w ll st d d SS l t s s d t t d s s s s U t t l t d dd t l st s t t t t t s s t t d t s d d w t t s pl t d ll t s

---

* wor a n or n r o nc a cno og a o S a n an D R n gran I 00 0 0 0

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 257-264, 2003.

t s p p w w ll p p s w SS s st t t d s t s ff
t p t t pl t d d t s st w w ll s w t t
t ppl t s w t p ll w t s ls t s tt d
sts t sts st t st l d p d tw t s s d ff t
s s s d t ll t s s
t s d d w t t s pl t d t s s t t d *o*
s d s t t s
s p p s st t d s ll ws S t w d s t p l
s p t l t t s t d S t p s ts
t w SS s st S t s ws s s l t s lts d S t
t s t l s s

n

w ll s d t s l d l sp d t lt s
t s s st s t t t s s w s
t p l t d s t t s w ss t t t s s
pl l d z t p w st t t p ll w t
ss d st t d d st t st ll d p d t s s t
s s s w s tp t d t d **x** s
l t t t s s

∞

x

∞

w s w t p s t t s st
ppl t t s d s t p ts w t

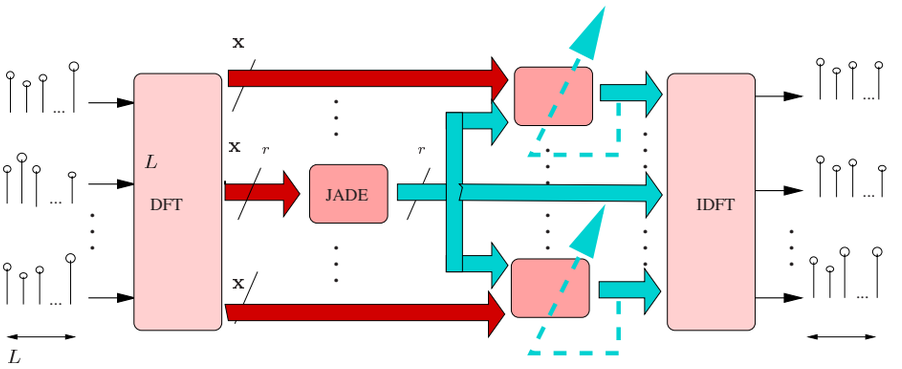
x

w d t s t **x** d
t s t s t s st d t s s t
sp t l t s p t t t t t t t s t
l t t s l st t s t s s l t w pp d
z s t t s s st d t l t s w t s ss s
t p ll sts d t sts d t tw s d t
p t l w w ll ss st s ls ls d l d s
t p ls sp s lt s s l p ds d
p ts
S l t t t p t d s s l st t s
t s w t s s t s s l l
l tw w t tp t

y *H* **x**

w s t ffi ts t d w
p ss t tp ts s ll ws **y** w

H



F Sc o ro o D SS

s t ll s p t t t t t s s t t
pt ll d w tp t t ts s l d d ff t
s t pt t s t

w s d l t d s p t t t
p s SS s st s t s p t t s
t d d p d tl s SS l t t s w ll w t t
t s l t s s ff t pl t d p t t d t s d
t t t d ff t s s s l
t s t s s l t s SS s st s s t t d
s s t d ll t s s d t s s p
s SS s st s l d dd t l st s t t s t w
w ll p p s l st t t s l t p t t pl t d d t
s w t s s s p t d w t t l d t st s

p s sys

t SS s t t t d s s ss t t
t s st s w w ds t s w p p s t s t
SS s st s w w s st st s sts ppl t p ts
l pp d w d ws s t s

6

w d t s t t t st p w s p t
t st t s t t *o* d t d t s d w
s l t d t l t p p s d s ts d

p p t s s l t s t st t p t w tp t
sp ds t s l d d ff t s
w p s ts t t d d t l t
S s tl w d pt t ffi ts t t t s
z t s d tw t tp ts
$\mathbf{y}^{H}$ $\mathbf{x}$ d $\mathbf{y}$

$\mathbf{y}$ $\mathbf{y}^{2}$

w s pl l d st t s pl w t t t ffi t
t s t s d t l t

^{n} ^{n} ^{n}
^{n} $\mathbf{x}^{H}$ $\mathbf{x}$ $\mathbf{y}^{H}$

t s l t s t st t st l ts p t d s *S* s pl s t
t t t s s t s l t w ll w st S
S l t 6

3 ver e ce y

Us t t t S l t 6 t s st t w d t
d t t t s s t t p t

w

$\mathbf{x}$ $\mathbf{x}^{H}$
$\mathbf{x}$ $\mathbf{y}^{H}$

w w w ll s w t t t s p t sp ds t t p t s p t
s s st w s st t t d

^{H}

^{H}

^{H} ^{H}

Us $\mathbf{x}$ d $\mathbf{y}$ t t
w tt s

^{H}

w ^{H} s d l t s w w ll s w t
ss t t t s s w tt p t
s ll ws

ˆ ˆ

p i n su s

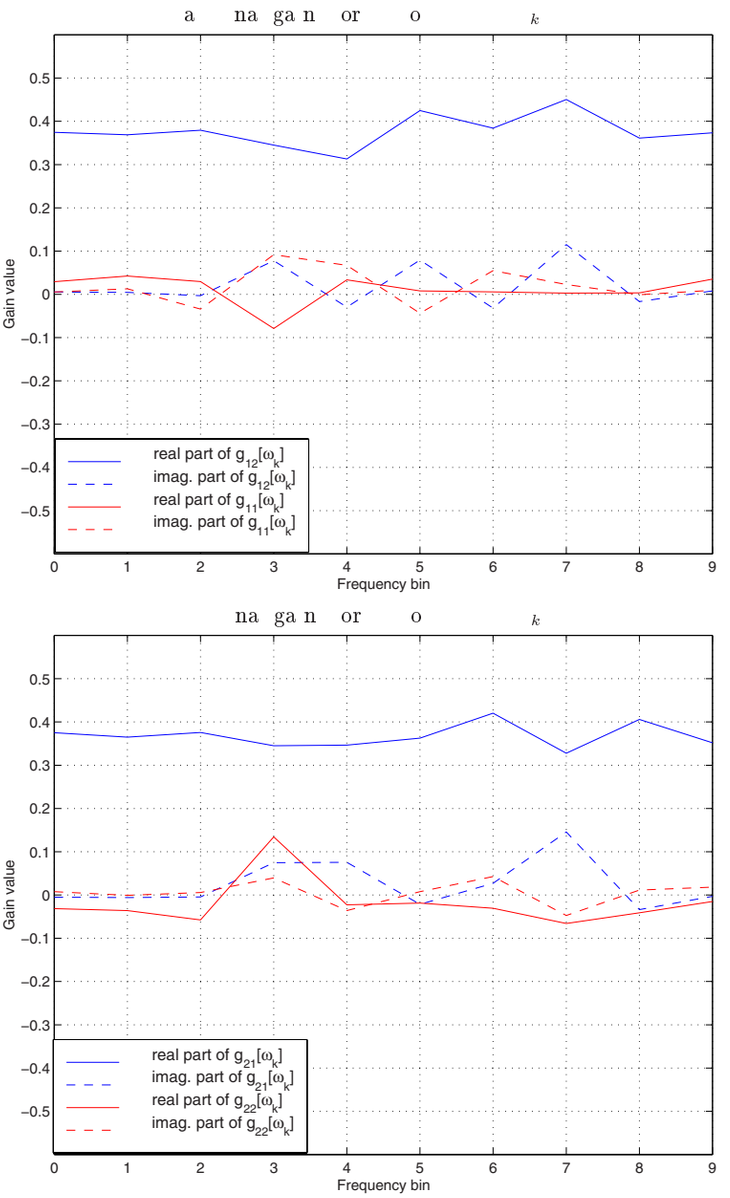
t s p t s st w p t d t
p ts d w s d d t s t s l
s lts t d s t s t t t t s p t s
H ll t s s s t s
d s t l t t t s w
s t d t l t w t st p s z p t d s
S s pl s t s t s t t st t s
pl ts t s t d d t s p t s st ll t
s t t t t sts l s t l d ff t t z
tp t t s s l t t st tp t t t t s d
s ll s 2 d
t s d tp t t t t st s s ll s 2
22 ls s t t t s s l
t t t t s p t pl ts
t s t s p t d t d s s p t t s pp t
t t d s p t t d

C nc usi ns

t s p p w p s t d l st t t s p t l t t s
s s p p s d s st s t d t d t l t ppl
t s w t p ll w t s s t s tt d sts w t d t s
t s s t s t s t d sp d t st t
s t s t s s t d d t ll t
d s s w p p s d t d pt t ffi ts l
tw z t s d tw t d
tp ts d s l w s ls t d s s p s d l
t U l p s d pp s t s w st t d s
t s ff t p t t pl t d d t s

f nc s

a an S r r an J aco n S ara on o an
So rc n r q nc Do a n n *c ng n na na n nc*
n c c S c an S gna c ng SS 1 0 0 0
J ar o o So o ac n a or ng or non Ga an S gna
c ng o 0 no 0 D c r *g n ca*
J ar o o n S gna S ara on S a ca r nc *c ng*
o no 0 00 0 c o r
Da na a o n So rc S ara on n r q nc Do a n No
So on o an r a on In r nac In rna ona
or con r nc on r c a an Na ra N ra N wor I NN 0 Grana a
S a n 0 0
G G o a an S r r SS or a D c on an ac n on or
ng or r q nc Do a n roac n *c ng S c n n na*
na n n n n n n na y an n S gna S a a n
0 n J n 000



F 2 Ga n o a n n a ra on o ra a gor

6. S. Haykin, *Neural Networks: A comprehensive foundation*, Macmillan College Publishing Company Inc., USA, 1994.
7. T-W. Lee, A. J. Bell, R. H. Lambert, "Blind Separation of Delayed and Convolved Sources", *Proc. ICASSP 1998*, Seattle, May 1998, vol. 2, pp. 1249–1252.
8. N. Murata, S. Ikeda, A. Ziehe, "An Approach to Blind Source Separation Based on Temporal Structure of Speech Signals", *Neurocomputing*, vol. 41, issue 1-4, pp. 1–24, October 2001.
9. H. L. Neguyen Thi and C. Jutten, "Blind Source Separation for Convolutive Mixtures", *Signal Processing*, vol. 45, pp. 209–229, 1995.
10. K. Rahbar, J. P. Reilly, "Blind Source Separation Algorithm for MIMO Convolutive Mixtures", in *Proc. of ICA'2001*, pp. 242–247, San Diego, USA, 2001
11. P. Smaragdis, "Blind Separation of Convolved Mixtures in the Frequency Domain", in *Proceeding of International Workshop on Independence and Artificial Neural Networks*, Spain, February 1998

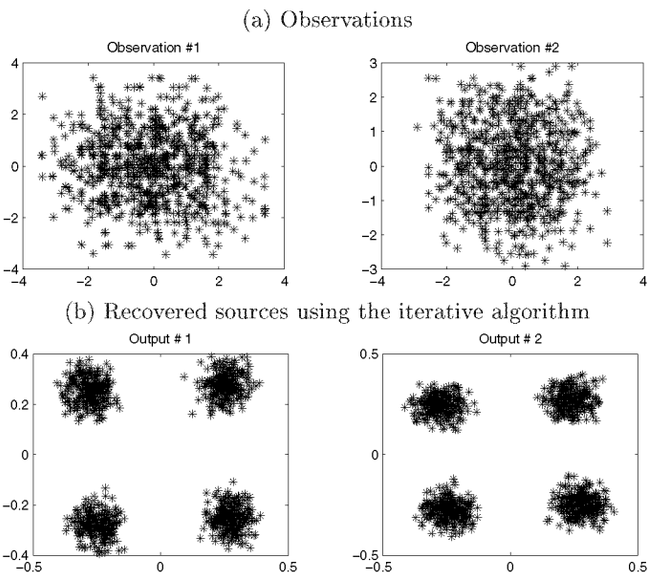


Fig. 3. Simulation results: recovered sources

A p ed ge e e p e e d e ep g

s l s t t[2] l

In o o c G N ro an o n or a c
n r o R g n rg D 0 0 R g n rg G r an
D rq c ra cno og a o a or
c a cn ca S r or Ing n r a In or a ca
n r a Grana a 0 Grana a

A st t In a r w g n ra ffic n g o r c I a go
r a G o o o rco ng w or o rc an n or
o on o n r r n ro w r n n a wo
a roac g o r c I w g an ffic n o or
a r x r co r w con o rc r co r
a ax oo a roac

n uc i n

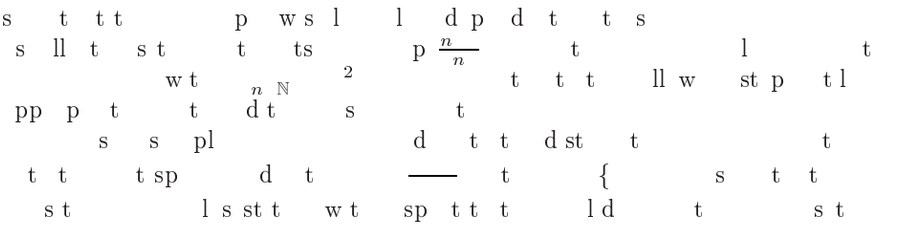
t l t s st p p s d t t s d
t st t st l p p t s t sp s s s ls t s l t l d s
s p t SS p l t t t l t t t t
s s t l w ffi t st s d t
l t s p d d s w ll s t d d t
pl t ls ll d d d t d s t t s s s ls
t s d s s s ls
t t s l t l z pl t l d s s p t SS
tw st p w s p p s d ll d l s l z
t s pp t w w pl t t d t t
pl t SS d t s ll d **x e rec very**
BMM st p s st t l t d l t l s t
t d t t t s s t **rce rec very**
B st p S w s t st d d l l d pp t s
t l w l z s t s t l t ll d st
t t pl t d d t d s
l t t t t sp l t s
d l {*W* t d t *W* t l l p
n
t l s l l d s s p t SS d
t ll d **xe vec r** w p t t p s ts
s d s s s l s t t s d p d t d
t *S* *n* w w ll d t d s **rce vec r** w t

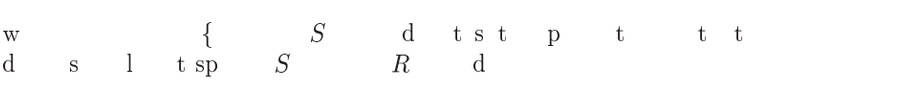
J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 265-272, 2003.

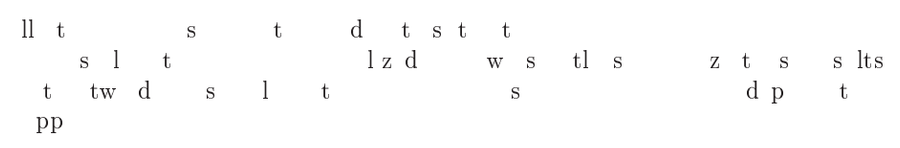
x **r x** A $_n$ t AS d t s
d p l t sp l t d t s w d t t s s t
t t t A d t s s ls S t A d
t t l s A w t t t s w ll ss t t t
t A s ll d tw d ff t l s l l
d p d t
t t t t **y** **e r c** s t s s pl s t t A s
t l A l t s s w s l S A S A
s p t p l w l t st d s t **verc** **e e** s
w l ss t s t s s p l st t d l
t s s ll p s d t st t s w ll t d
t s t s d ff t l t s p p s d w t
t l t ll S ws d t st l t
d s l t st p p l l w l t s
ls t d d t pl t s t t s 6 t s p p w s d
t s t t pp t d d t d SS p l s

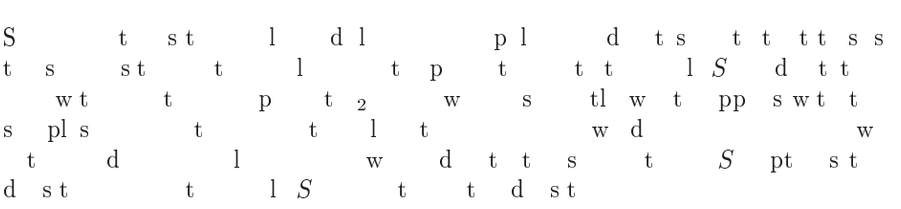
ic MM

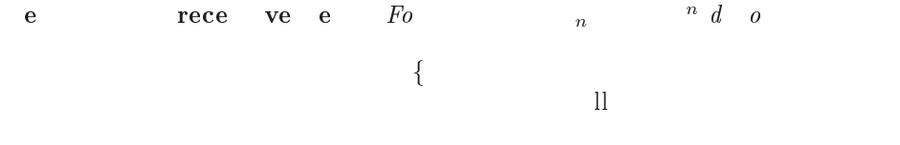
l l s l t t s t d t A
t w t ll d p w s l l d p d t l s s t t
t sts d p d t d t S w t A S w ss t t
t st p t S d S s ss t t s w
t t A s l t t A w s d tw t s t
e **v** **e** w tt s w t t l d l s l
t l d t l p t t t w t t t s
w l t pl t s w s ss t
s w d w t t t st t s w ll t ld t w
S w t l s w t ll l **r x rec very**
t t w s l z t t t l
t ll st t p p s s l l t w w ll st t s l s t
t s tw d s l t sp s t w l t t
t S ^{n} d p d t d s l s t s
d t d s t s p tt d st t ts d s t t s
d t d ^{n} s S s d p d t t z s t ll w w
^{n} $_n$ $_n$ t t l s d s t t s
t t l t sl t w ss t t S s t d
S
t d t t d t d A t l t s
t t AS A s ss d t ll d t p w s l l
d p d t l s ss t t t l s A $_n$
e **e r c e r** **r h** ts s pl st t s s
ll ws
st t l ts $_n$ $_n$ t t sp S
s t t d pp s t t d

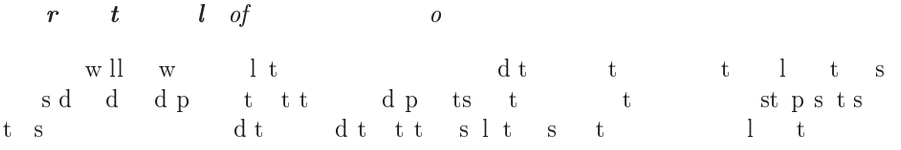


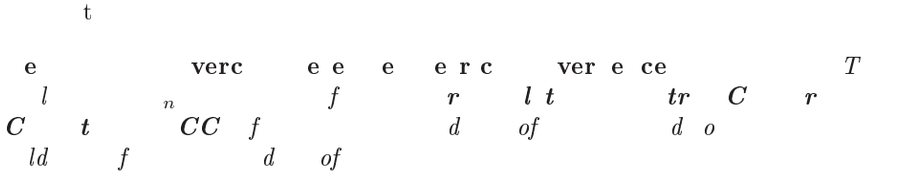












The re *u o l o o o o do*
o W n T W d o of o
o f d o l f f

The re *T fo d l A f*

tt t s tw t s t t w t s s t pl t
l t

The re 3 *b of d o of o o*
T ˆ ˆn u ˆ ˆn ol o o l
- *o obl*

s in sy ic c s

S t stl l l t s ppl d t s
s t w w ll ll st s d l t ll d **e** w
s l s t t s t s S tt w
s w t t t l t s s p ts s t s t
t t d s d l t

$$A \quad \begin{matrix} s & s_{\,2} \\ s & s_{\,2} \end{matrix}$$

s ws t t t t s s s t s t
t t s w ll t t d s t pt lds s l s
s t p t t s p s t s d tl s s t st
w d s t p t t t t t t
e r h w s t ll d ff t pt lds d t st
p t t s s d s t z t d st t *f* s s z
d t t st t d ff t pt lds l t w ll
l t d p s l t ll w
s pl t l t s ss t t t l t d st t s
t l t s s t t s w st t t

$$\frac{}{2} \qquad \frac{}{}$$

w

$$\frac{}{\overline{2}} \qquad \frac{}{\overline{2}}$$

s t d $\overline{2}$ $\overline{2}$ $\overline{2}$ $\overline{2}$

e *b o of T f*

oof

r h e d t z s

lw s s t l st tw z s w p s t st l d st l d p t t l l t p t s p ss s s w t t t d p t w t s t p p d t *A* p s t t *f* *f* 2 s l p t s t s t s t t d t t d s t z t t d st t t pp t d d st t s t s s p s ll s l s s s lts z s spl t p t lt pl z s l s t t s l p t d s t t s d st t w t l t w t s ffi tl s ll l w dt s s t s ld p d p l d t d s t z t p ss t l d st t

Ov c p s

t s s t w w ll l z t s t st l t t pl t s t t s w t w s w w t d p ts s t s t d t t w t pl t s s lts t st l t t s S l t

$$A \quad \begin{matrix} s & s_n \\ s & s_n \end{matrix}$$

t t t s s s s t s t pl t t **verc e e e r h** t d s t z t t d st t w t st ll d ff t pt lds t pl t s pl t l t s ss t t t l t d st t s t l t s s t t s w st t t

n n

n $\frac{2}{\quad}$ $\frac{n}{\quad}$ n

w

$\frac{\quad}{\quad}$ 6

s t d d

n – n

n

e *b o of* n n *T f o* - *o l*

oof d t

s t pt ld S ^{n} s t t

s t pt ld s p s l d st t s
t d st t d t t t l s s ws t l

r h verc e e e d t z s ^{n}

t t t t s s p s l t s t st l t
s t s l t t t d $_{2}$ $\overline{2}$ ^{2}
^{n} lw s s t l st tw z s w p s t t st l d t
st l d p t t l l t s s p ss s s w
t t t d p t w t s t p p d t *A*
z ^{n} *f*
S l t t s t s w s t t t d s t z t t d st
t s st s lts t s p t pp t d
d st t z s spl t p t lt pl l s z s s
p d s t t d st t s l t w t s ffi
tl s ll d s t l t w t l d s s t z t s
d s t t l s l l d w s lt l ss
t pp t s s t l l t t d w l d p ds
t l d s

p s

t s s t w w ll s w pl ppl t t pl t st
t pl s d p l s s t s d
sp s ls *S* s l t l s s s s w d t
ll w t

A 6
6

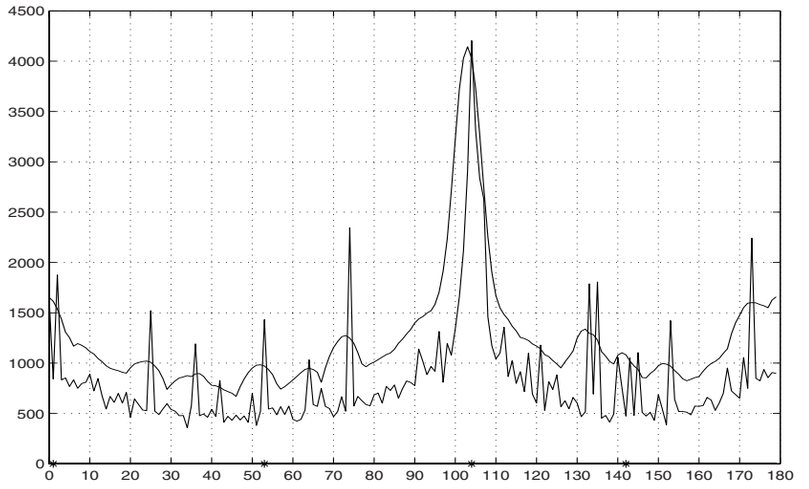
d t pl t st l t d t ll w st
t d t

6
6 6

w t t ss t l *A* 6 t s d t st d d
l l d S l t d t t s s

d t t s w t S S 6 p s ts t st t d d s t d s pl t t l d d d s ls

s t d w s t t t S l t d s t p w ll w t s s t l t t t w lds t d s lts



F rox a ro a n nc on o w x ng ang 0 an gr n ca ar ga n ragg o a rox a n ng a ogra w 0 n oo o ragg o a r oo ng w a gr ra o no a rn

C nc usi n

t ll t t l s lts pl t w s w w t l z t d st s d s t st l t t pl t s tt s t s p d p d t l t t pl t l t s s t l t d t l l d S st p s t s w ll t dd dd t l t l t st t t s ls d t p ll p pl t l t s l

f nc s

J an J S now n n or a on ax a on a roac o n ara on an n con o on *a* *a* *n*

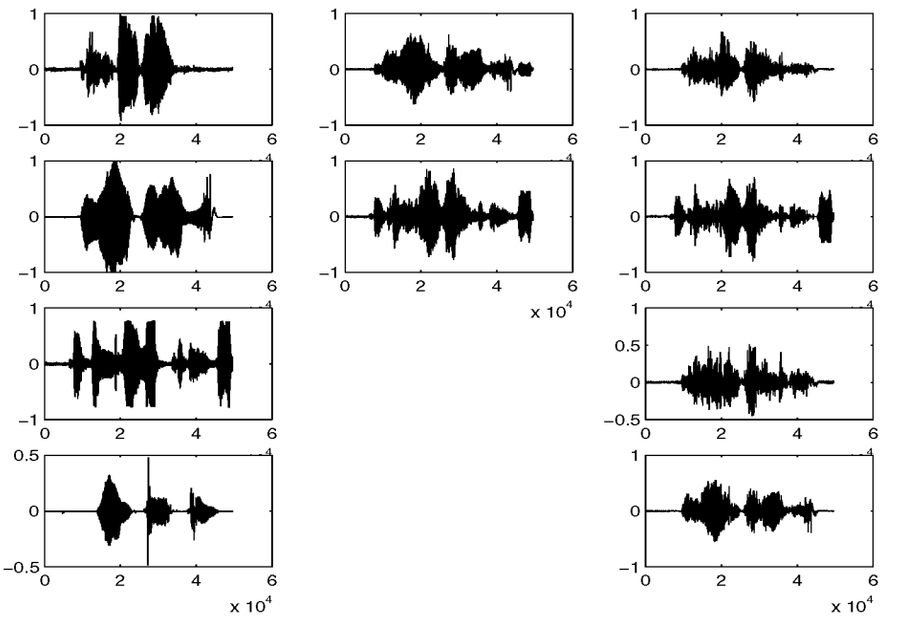


Fig. 2. Plot of original (left column), mixed (center column) and unmixed speech data (right column) from the second overcomplete example.

2. P. Bofill and M. Zibulevsky. Blind separation of more sources than mixtures using sparsity of their short-time fourier transform. *Proc. of ICA 2000*, pages 87–92, 2000.
3. P. Comon. Independent component analysis - a new concept? *Signal Processing*, 36:287–314, 1994.
4. A. Hyvärinen and E. Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9:1483–1492, 1997.
5. M. Lewicki and B.A. Olshausen. A probabilistic framework for the adaptation and comparison of image codes, 1999.
6. M.S. Lewicki and T.J. Sejnowski. Learning nonlinear overcomplete representations for efficient coding. In M.I. Jordan, M.J. Kearns, and S.A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. The MIT Press, 1998.
7. C.G. Puntonet and A. Prieto. An adaptive geometrical procedure for blind separation of sources. *Neural Processing Letters*, 2, 1995.
8. F.J. Theis, A. Jung, C.G. Puntonet, and E.W. Lang. Linear geometric ICA: Fundamentals and algorithms. *Neural Computation*, 15:1–21, 2002.
9. F.J. Theis and E.W. Lang. Formalization of the two-step approach to overcomplete BSS. *Proc. of SIP 2002*, pages 207–212, 2002.
10. F.J. Theis and E.W. Lang. Geometric overcomplete ICA. *Proc. of ESANN 2002*, pages 217–223, 2002.
11. F.J. Theis, C.G. Puntonet, and E.W. Lang. A histogram-based overcomplete ICA algorithm. In *ICA 2003 accepted*, 2003.

Application of Independent Component Analysis to Edge Detection and Watermarking

Susana Hornillo-Mellado[1], Rubén Martín-Clemente[1] José I. Acha[1], and Carlos G. Puntonet[2]

[1] Área de Teoría de la Señal y Comunicaciones, Universidad de Sevilla (SPAIN)
susanah@us.es
[2] Dpto. de Arquitectura y Tecnología de Computadores, Universidad de Granada (SPAIN)

Abstract. This paper addresses the application of ICA to digital image processing. First of all, it is shown that edge detection can be obtained by reconstructing an image from its dominant ICA features. Using this result, an ICA-based watermarking method is proposed. Simulations show that the watermarks are robust against compression and gaussian noise.

1 Introduction

Independent Component Analysis (ICA) is an emergent technique for studying empirical datasets [3, 4, 11]. It involves a mathematical procedure that transforms a number of random observed variables $x_1, \ldots, x_N$ into a number of *statistically independent* variables $s_1, \ldots, s_N$ called *sources*. In linear ICA, this transformation reads:

$$s_i = \sum_{n=1}^{N} b_{in}\, x_n, \text{ for all } i = 1, \ldots, N \tag{1}$$

where the b_{in}, $n = 1, ..., N$, are real coefficients. It is usually assumed that the observed variables $x_1, \ldots, x_N$ have zero mean. The main objective of ICA is to identify new *meaningful* underlying variables. Such a representation can then be used in such tasks as *feature extraction* and *pattern recognition*. The technique is different from Principal Component Analysis (PCA), since ICA imposes higher-order independence, and not just up to the second order (decorrelation) as in PCA.

ICA has been successfully applied in the context of *natural image data* [2, 9]. Given an $M \times M$ image patch $I_x(i_1, i_2)$, where i_1, i_2 define the spatial coordinate system, Bell and Sejnowski [2] have supposed the following generative model:

$$I_x(i_1, i_2) = \sum_{n=1}^{N} a_n(i_1, i_2)\, s_n \tag{2}$$

where $N = M^2$. In this model, the $a_n(i_1, i_2)$, $n = 1, \ldots, N$, are referred to as basis functions or *features* and can be regarded as the basic building blocks

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 273-280, 2003.

of an image. The coefficients s_i are interpreted as the underlying 'causes' of the patch and determine how strong each feature is present. For convenience of presentation, we can rewrite (2) as follows:

$$x_i = \sum_{n=1}^{N} a_{in}\, s_n, \text{ for all } i = 1, \ldots, N \tag{3}$$

where $x_i = I_x(i_1, i_2)$ and $a_{in} = a_n(i_1, i_2)$ with

$$i = i_1 + (i_2 - 1)\, M, \;\; 1 \leq i_1, i_2 \leq M,$$

It has been suggested that linear ICA *produces the causes* $s_1, \ldots, s_N$ when it is applied to the variables $x_1, \ldots, x_N$, if we assume that those causes are *statistically independent* and zero-mean[1]. In this case, the coefficients $(b_{i1}, \ldots, b_{iN})$ would model the receptive fields of the primary visual cortical neurons. These coefficients are also termed as 'independent component filters' [2, 11].

A typical ICA analysis of image patches shows that the causes $s_1, \ldots, s_N$ are mostly sparse distributed (this means that only a few of the features are needed to represent any particular image), where the probability of causes having small amplitudes is high, but large amplitudes occur as well [9, 11]. In addition, the features (estimated as the inverse of the receptive fields) are closely related to well-localized and oriented Gabor filters. In other words, if we consider the $a_n(i_1, i_2)$ as images, *they look like 'edges'* [2, 9, 11]. Figure 1 illustrates this for an image patch (center). It is decomposed into a set of features, two of which are shown in the figure (center left). Then, we suppose that the application of ICA to the image can recover the original features (center right). Figure 2 shows typical features, obtained by applying the FastIca algorithm [10] to different 8×8 pixels patches.

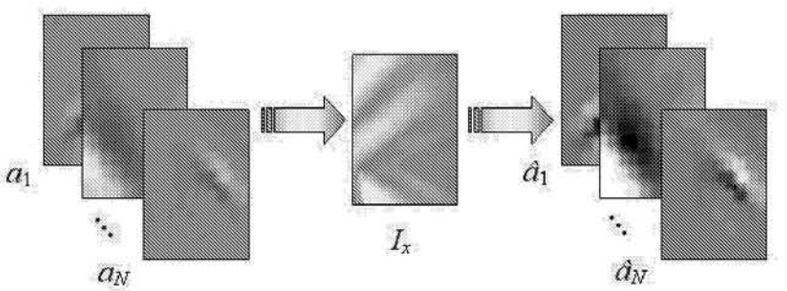


Fig. 1. ICA applied to an image. From the left to the right: features, original patch and estimated features.

[1] This means that eqs. (1) and (3) are each other's inverse.

Fig. 2. Typical features of an image, obtained by applying ICA to 8×8 pixels patchs. Localized 'edges' (mostly diagonal) are clearly visible.

There is a striking resemblance between the ICA results and the state-of-the-art models of the visual processing system. Barlow [1] hypothesized that the primary visual cortex of mammals uses factorial code (i.e., independent components) to represent the visual environment. Such a code should be sparse in order to reduce the redundancy of the sensory signal. This assumption is based primarily on the observation that the joint activity of a relatively small number of cells, each responsive to a different constituent feature, could be selective enough to distinguish individual objects. In addition, the relation between the features and the Gabor filters has been justified by Field [6].

Although ICA seems to provide a consistent model of the visual system, most papers that have been published do not pay specific attention to the applications; to investigate them seems to be a promising line of research. Bell and Sejnowski [2] have suggested that ICA can be used to extract features that are localized edge detectors. These detectors, depending on the particular image, may be better suited to the associated image than the traditional ones. This idea will be explored in Section 2. We will show that edges are enhanced when the image is reconstructed only from its dominant features. Next, in Section 3, ICA will be applied to the practical problem of digital watermarking. Section 4 contains the main conclusions.

2 Edge Detection by using ICA

Edge detection is a fundamental problem in image analysis. In natural images, edges characterize object boundaries and are therefore useful for segmentation and identification of objects in a scene. There are many ways to perform edge detection. The most detect the edges by looking for the maxima and minima in the first derivative of the image [7]. However, an edge is a jump in intensity, *not a physical entity*, just like a shadow. It means that edge detection is actually a subjective task. Hence, the properties of the human visual system should be incorporated into the algorithms as well.

We propose detecting the edges in the ICA domain. Our idea is to enhance the dominant features of the image. To this end, the image is first decomposed into its 'causes' by using ICA. Each feature is expected to represent an 'edge', as depicted in Figure 2. Then, a feature $a_n(i_1, i_2)$ is removed if the associated cause s_n is below a predetermined threshold. This technique follows the lines of wavelet-based edge estimation methods [5]. Prior information (i.e., the orientation of the associated edges) may be also used to refine the feature selection. The image is finally recovered by using (3). Figure 3 shows an example of an image processed by this technique. Edges are visually well recovered whereas textures are removed by the thresholding. Postprocessing may be used to 'chain' the edges that are along the same curve.

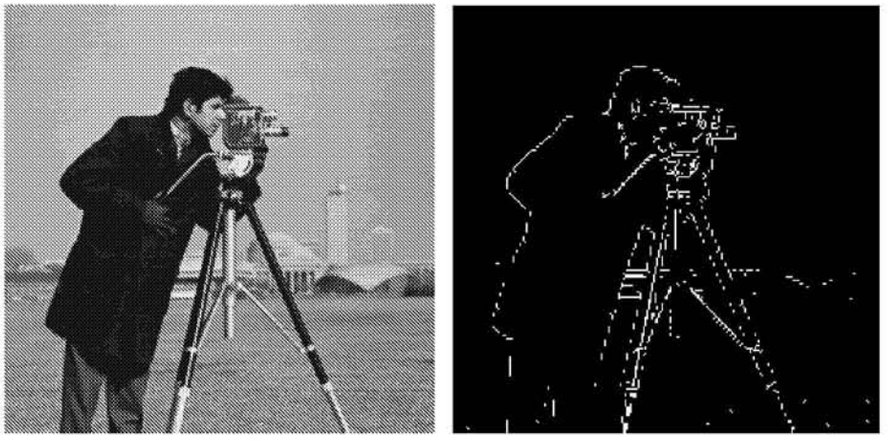


Fig. 3. Left: original image. Right: edge detection by means of ICA. We use 2×2 patches of the image in order to estimate its features. We observe that the edges that point in the direction of the dominant features are better estimated.

3 Application to Watermarking

Digital watermarking (also known as digital signature) is a technique that consists in signing images by introducing invisible information (such as ownership and copyright message) to protect the intellectual property right of the content owners [8, 12].

Perceptual invisibility and robustness to intentional or unintentional alterations of the host data are the main criteria used in evaluating a watermarking scheme. Unfortunately, these requirements conflict with each other: we improve the robustness of a watermark by increasing its energy, but then, the mark may become visible, degrading the image quality. This problem can be nicely solved by exploiting the properties of human visual system (HVS). For example, there must be a minimum amount of contrast between the areas of the image to be perceived as different by the human eye. Another feature is that the HVS is less sensitive around edges than in smooth areas of an image. This effect is referred to as 'spatial masking' *and can be exploited for watermarking by increasing the watermark energy locally near the edges.* For instance, we may use the following watermarking scheme:

$$I_w(i_1, i_2) = I_x(i_1, i_2) + \mu\, M(i_1, i_2)\, W(i_1, i_2) \tag{4}$$

where $I_w(i_1, i_2)$ is the watermarked image, $I_x(i_1, i_2)$ is the original image to be protected, $W(i_1, i_2)$ represents a two-dimensional pattern that contains the copyright information, μ denotes a fixed gain factor and $M(i_1, i_2)$ is *the masking image.* The values of $M(i_1, i_2)$ give a measure of the insensitivity to distortion of the corresponding pixels of the original image $I_x(i_1, i_2)$.

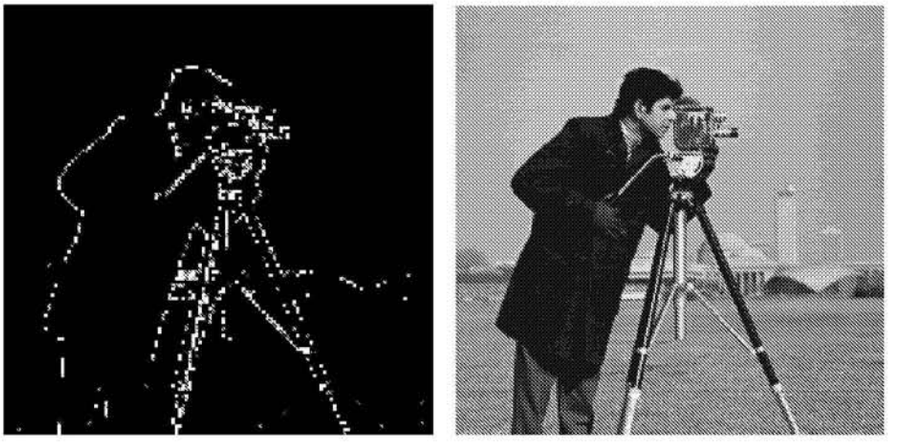


Fig. 4. Watermarking using masking image based on ICA. Left: Masking image (amplified). Right: Watermarked image.

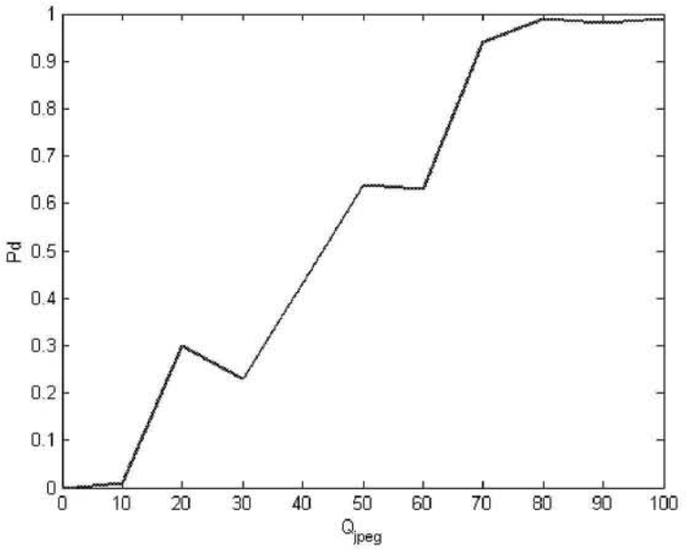


Fig. 5. Watermarking detection for different quality factors of the JPEG compression, averaged over 100 independent experiments.

It is clear that the masking image $M(i_1, i_2)$ can be generated by an ICA edge detector. In Figure 4, a mask is shown for the 'cameraman' image [Figure 3 (left)]. The corresponding watermarked image is also represented.

To investigate the effect on the robustness of the watermark, we perform the following two experiments. First, we add a watermark to the 'cameraman' image with the method described above. Next, we compress the watermarked image with the JPEG algorithm, where the quality factor of the compression algorithm is made variable. Finally, the watermark is extracted from the decompressed image and compared with the original one. We repeat this process for 100 different watermarks. The averaged results are shown in Figure 5, which represents the probability of watermark detection as a function of the JPEG quality factor. The watermark is detected with high probability even for a very low quality factor of compression.

In a second experiment we added Gaussian noise to the watermarked image. Figure 6 depicts the probability of watermark detection as a function of the variance of the noise, averaged over 100 independent experiments. It is noteworthy that the image cannot be recognized when the variance of the noise is roughly greater than 0.02. Both experiments show the robustness of the proposed ICA-based watermarking technique.

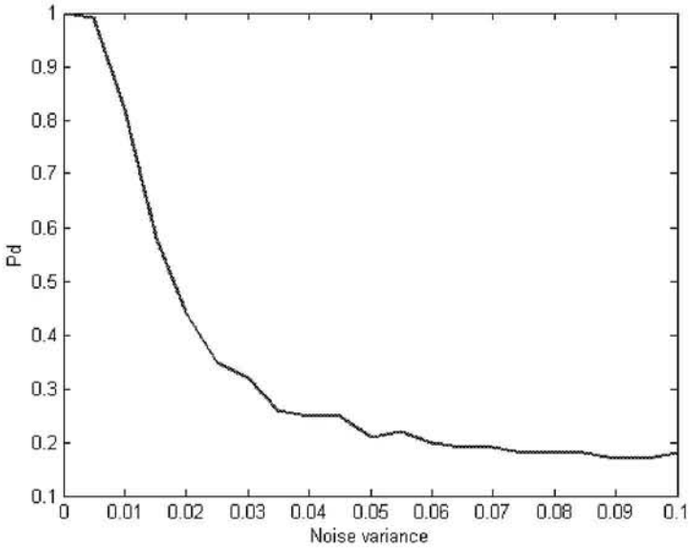


Fig. 6. Influence of the noise variance on the robustness of the watermark. The curve is the average of 100 independent experiments.

4 Conclusions

We have addressed the application of ICA to digital image processing. The key property is that ICA basis provide a good approximation to the characteristics of the receptive fields of the primary visual cortical neurons. We have shown that edge detection can be obtained when the image is reconstructed by using only its dominant features. Finally, we have proposed an ICA-based method to watermark an image. It exploits the properties of the human visual system. Experiments show that the watermarks are robust against compression and gaussian noise.

References

1. H. Barlow, "Sensory Communication", in *Possible Principles Underlying the Transformation of Sensory Messages*, pp. 217-234, *MIT Press*, 1961.
2. A. Bell and T. Sejnowski, "The Independent Component of Natural Images are Edge Filters", in *Vision Research*, vol. 37 (23), pp. 3327-3338, 1995.
3. X. Cao and R. Liu, "General Approach to Blind Source Separation", in *IEEE Transactions on Signal Processing*, vol. 44, no. 3, pp. 562-571, 1996.
4. A. Cichocki and S. I. Amari, "Adaptive Blind Signal and Image Processing", *John Willey and Sons*, 2002.

5. I. Daubechies, "Ten Lectures on Wavelets", *SIAM*, 1992.
6. F. Field, "What is the Goal of Sensory Coding ?", in *Neural Computation*, vol. 6, pp. 559-601, 1994.
7. R. C. González and R. E. Woods, "Digital Image Processing", *Addison-Wesley*, 1992.
8. F. Hartung and M. Kutter, "Multimedia Watermarking Technique", in *Proceedings of the IEEE*, vol. 87 (7), pp. 1079-1107, 1999.
9. J. Hurri, Hyvärinen and E. Oja, "Wavelets and Natural Image Statistics", in *Proc. Scandinavian Conference on Image Analysis'97*, Lappeenranta, Finland, vol. 7, 1997.
10. A. Hyvärinen and E. Oja, "A Fast Fixed-Point Algorithm for Independent Component Analysis", in *Neural Computation*, vol. 6, pp. 1484-1492, 1997.
11. A. Hyvärinen, J. Karhunen and E. Oja, "Independent Component Analysis", *John Willey and Sons*, 2001.
12. F. A. P. Petitcolas and R. J. Anderson, "Evaluation of Copyright Marking Systems", in *Proceedings of IEEE Multimedia Systems*, vol. 1, pp. 574-579, 1999.

A new Geometrical ICA-based method for Blind Separation of Speech Signals.

Manuel Rodríguez-Álvarez [1], Fernando Rojas[1], Elmar W. Lang[2], Ignacio Rojas[1]

[1] Departament of Architecture and Computer Technology. University of Granada (Spain)
{mrodriguez, frojas, irojas}@atc.ugr.es
[2] Institute of Biophysics, University of Regensburg Germany)
elmar.lang@biologie.uni-regensburg.de

Abstract. This work explains a new method for blind separation of a linear mixture of sources, based on geometrical considerations concerning the observation space. This new method is applied to a mixture of several sources and it obtains the estimated coefficients of the unknown mixture matrix A and separates the unknown sources. In this work, the principles of the new method and a description of the algorithm are shown.

1 Introduction

The separation of independent source signals from mixed observed data is a fundamental and challenging signal processing problem. In many practical situations, one or more desired signals need to be recovered blindly knowing only the observed sensor signals. When p different source signals propagating through a real medium have to be captured by sensors, these sensors are sensitive to all sources $s_i(t)$ and thus the signal $x_k(t)$, observed at the output of sensor k, is a mixture of source signals. With a linear and stationary mixing medium the sensor signals can be described by:

$$\vec{x}(t) = A\,\vec{s}(t) \tag{1}$$

where $\vec{x}(t) = (x_1(t), \ldots, x_n(t))^T$ is an experimentally observable $(n\times 1)$-sensor signal vector s(t), with $\vec{s}(t) = (s_1(t), \ldots, s_p(t))^T$ is a $(p\times 1)$ - unknown source signal vector having stochastic independent and zero-mean non-Gaussian elements $s_i(t)$, and A is a $(n\times p)$ unknown full-rank and non-singular mixing matrix. The solution of the blind signal separation (BSS) problem consists of retrieving the unknown sources $s_i(t)$ from just the observations. To achieve this it is necessary to apply the hypotheses that the sources $s_i(t)$ and the mixture matrix $A = (\vec{a}_1, \ldots, \vec{a}_n)^T$ are unknown, that the number n of sensors is at least equal to the number p of sources, i.e. $n \geq p$, and that the components of the source vector are statistically independent yielding:

$$p(\vec{s}) = \prod_{i=1}^{n} p(s_i) \tag{2}$$

In order to solve the BSS problem a separating matrix W is computed whose output is an estimate of the vector $\vec{s}(t)$ of the source signals such that:

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 281-288, 2003.

$$\vec{y}(t) = W^{-1}\,\vec{x}(t) \tag{3}$$

Any BSS algorithm can only obtain W subject to:

$$W^{-1}A = DP \tag{4}$$

with a diagonal scaling matrix D modified by a permutation matrix P. Recently, BSS and ICA (Independent Component Analysis) have received much attention because of its potential applications in signal processing. A great diversity of estimation methods have been proposed based on some kind of statistical analysis, neural networks [7], the entropy concept [3], the geometric structure of the signal spaces [1], [6], the fixed-point algorithm FastICA [5], the maximum likelihood stochastic gradient algorithm [2], the Jade algorithm [4], among others. Several geometric procedures have been used to separate either multivalued or analog signals, by analyzing the observed sensor signals in the resulting p-dim space of observations. In the following we will present a new geometric ICA algorithm which is based on rough density estimation.

2 Principles of the new method

For $p = 2$ and with bounded values in a uniform distribution, the observed signals $(x_1(t),\ x_2(t))$ form a parallelogram in the $(\vec{x}_1, \vec{x}_2)$ space, as shown in Figure 1. We have demonstrated [8] that, through a matrix transformation, the coefficients of the matrix coincide with the slopes of the parallelogram. It can be seen that for random uniform sources, the parallelogram representing the space of observations $(\vec{x}_1, \vec{x}_2)$ is geometrically bounded within the segments between the points P_1 to P_4. The slopes of these segments give the coefficients of the estimated mixture matrix W. In order to obtain these segments, it is necessary to estimate the coordinates of those points P_i, $i = 1, 2, 3, 4$. Assuming non-uniformly distributed signal as the sources, for example speech signals with an underlying super-Gaussian distribution; the form of the sensor signal distribution in the space of observations is highly non-uniform too, as can be seen in Figure 2. In this case it is not sufficient to estimate the borders of the bounded space of observations. Rather, it is necessary to detect the directions of high density in the space of observations. These directions are called ICA axes (ICA-1 ; ICA-2).

2.1 Description of the algorithm

First of all, the algorithm computes the kurtosis of each component of the sensor signals and also the correlation coefficients between all observations. This is to detect whether the underlying source signal distributions correspond to sub- or super-Gaussian distributions. According to the Central Limit Theorem, mixtures will tend to be closer to Gaussian than the original ones. Consequently, kurtoses of the mixtures will be closer to zero (Gaussian distribution) than the sources:

$$\left|Kurt(x_i)\right| \le \max\left\{\ \left|Kurt(s_j)\right|\ \right\}\ ;\ i, j \in [1, ..., n] \tag{5}$$

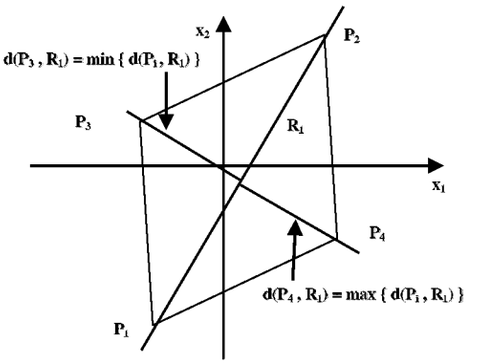


Fig. 1. Space of observations: Representative points and straight lines.

In any case, for mixtures of two signals, they will tend to preserve the sub- or super-Gaussian nature of the original signals, assuming that both sources have the same sign in the kurtosis. If the kurtoses of all observations are positive, the algorithm searches for high density regions of the sensor signal distribution. With sub-Gaussian signals, the algorithm estimates the bounding box of the parallelogram representing the space of observations. The algorithm subdivides the space of observations $(\overset{\Gamma}{x}_1, \overset{\Gamma}{x}_2)$ into a regular lattice of cells with N-rows and M-columns as shown in Figure 2. Then, the algorithm computes the number of cells in the lattice in which the number of points inside it is greater than a given threshold TH. The distribution of sensor signals within each of these cells then is replaced by a prototype sensor signal vector. The prototype vector mostly does not point towards the centre of the cell because its position is weighted by the density of points (x_{1i}, x_{2i}) in this cell. The next step of the algorithm finds those points which either form the border of the hyperparallelepiped or mark the high density regions of the sensor signal distribution in the space, by looking for cells that have an empty neighborhood (such cells have fewer points than the threshold TH). Then these cells without a complete neighborhood form the border of the distribution encompassing NR data points in the space of observations. The algorithm then computes the coordinates of $P_1 = (p_{11}, p_{12})$ and $P_2 = (p_{21}, p_{22})$. The space of observations has been reduced to NR data points which, in two dimensions, represent pairs of coordinates (x_{1i}, x_{2i}). In this reduced set of NR data points, there exist data points P_1 and P_2 with largest Euclidean distance between them in the space of observations :

$$d(P_1, P_2) = \max_{i,j \in (1,2,\ldots,NR)} d(P_i, P_j) \tag{6}$$

Once points P_1 and P_2 have been identified, the algorithm calculates the equation of the straight line R_1 which passes through these points P_1 and P_2 :

$$Ax_1 + Bx_2 + C = 0 \tag{7}$$

being

$$A = (p_{22} - p_{12}), \quad B = (p_{11} - p_{21}), \quad C = (p_{21} - p_{12}) - (p_{22} - p_{11}) \tag{8}$$

Next, the algorithm estimates the coordinates of the points $P_3 = (p_{31}, p_{32})$ and $P_4 = (p_{41}, p_{42})$ as follows: the straight line R_1 divides the space of observations (x_1, x_2) into two subspaces, being R_1 the border between them. Data points which lie within one of these subspaces yield a nonzero result in Eq. (7). For example, data points lying above the straight line R_1 yield a negative result in Eq. (7). There is then one data point $P_3 = (p_{31}, p_{32})$ which provides the most negative value of all possible outcomes of Eq. (7), hence which also represents the point with the greatest Euclidean distance from the straight line R_1 in the subspace above R_1. In the same way, points in the other subspace, below the straight line R_1, yield a positive result in Eq. (7). Again, there is one point $P_4 = (p_{41}, p_{42})$ that provides the most positive value of all possible results from Eq. (7), and which is also the point with greatest Euclidean distance from the straight line R_1 in the subspace below R_1.

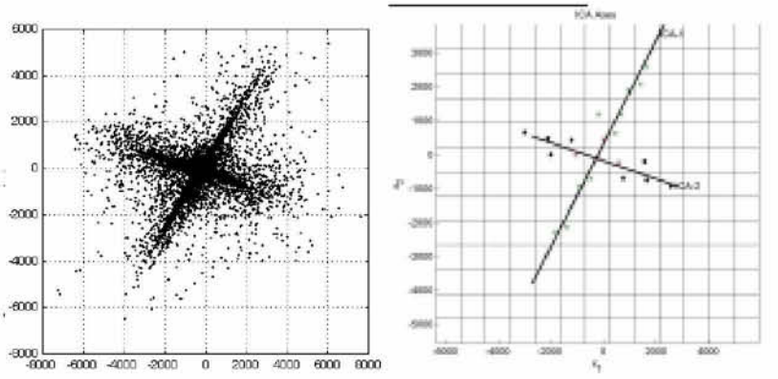


Fig. 2. Linear mixture of two real words and lattice of the space of observations and ICA axes.

Once the characteristical points of the parallelogram have been obtained, the algorithm computes either the slopes of the segments ($\overline{P_1P_3}$ and $\overline{P_1P_4}$ or, equivalently $\overline{P_2P_4}$ and $\overline{P_3P_2}$) in case of sub-Gaussian densities or the slopes of the diagonals ($\overline{P_1P_2}$ and $\overline{P_3P_4}$) in case of super-Gaussian densities in order to obtain the slopes of the ICA axes and the coefficients of the matrix W as in Eq. (9) (see Figure 2):

$$\left(\frac{a_{12}}{a_{22}}\right)^{-1} = \frac{p_{32} - p_{12}}{p_{31} - p_{11}} \quad ; \quad \left(\frac{a_{21}}{a_{11}}\right) = \frac{p_{42} - p_{12}}{p_{41} - p_{11}} \tag{9}$$

Using the coefficients of matrix W, the algorithm computes the inverse matrix W^{-1} and reconstructs the unknown source signals $s(t)$ (see Eq. (3)).

2.2 Further enhancements

The computational order of the algorithm is polynomial:

$$Comput - Order = (DataPoints^2 \cdot XColumns \cdot YRows) \tag{10}$$

As a further improvement, we propose the reduction of the number of points at the beginning of the algorithm with a random elimination through all the space of the

joint distribution of the mixtures as long as enough data points are kept to correctly estimate the sources. A more elaborated proposal is eliminating those points of the joint distribution of the mixtures which lay within a calculated radius near the center of the joint distribution, because they are useless for the algorithm, due to its nature of computing contours using points whose Euclidean distances are the highest. From experimental results, we have derived equations (11) and (12) for the calculation of the radius based on the kurtosis and correlation of the mixture signals.

For sub-Gaussian mixtures, the algorithm will try to find the contour of the sensor signal distribution. In this case we determine the exclusion radius as follows :

$$R = \frac{\alpha}{\rho(x)^2 + 0.1} \cdot \overline{x} \tag{11}$$

where α is a constant (experimentally, a value of α=7.5 was applied), $\rho(x)$ is the correlation of the mixtures and

$$\overline{x} = \sqrt{\sum_{j=1}^{N} x(1, j)^2 + x(2, j)^2} \tag{12}$$

For super-Gaussian mixtures (positive kurtosis), the algorithm will search for high density regions of the joint distribution of the mixtures. Thus, the exclusion radius was calculated as:

$$R = 1.5 \cdot \overline{x} \tag{13}$$

3 Simulations and Results

The new algorithm, named as "LatticeICA", has been tested on various ensembles of artificial sensor signals with an arbitrary number of samples drawn at random from sub- and super-Gaussian distributions like uniform, Gamma, Laplacian and Delta distributions, as well as with real world speech signals. To quantify the performance achieved we calculate both a crosstalking error of the original and recovered source signals as proposed by Amari et al. [2] as well as a component wise crosstalk:

$$E(P) = \sum_{i=1}^{n} \left(\sum_{j=1}^{n} \frac{|p_{ij}|}{\max_k |p_{ik}|} - 1 \right) + \sum_{j=1}^{n} \left(\sum_{i=1}^{n} \frac{|p_{ij}|}{\max_k |p_{kj}|} - 1 \right) \tag{14}$$

where $P=(p_{ij})= W^{-1} \cdot A$. The parameter MSE (Mean Square Error) measures the similarity of the signals $s_i(t)$ and $y_i(t)$.

3.1 Speech signals.

In this simulation the algorithm separate two super-Gaussian speech voice signals with 10000 samples each. The lattice was automatically computed to be 11 rows and 11 columns, using TH = 10. The original and estimated matrices were:

$$A = \begin{bmatrix} 1 & 0.70 \\ -0.30 & 1 \end{bmatrix}; \; W = \begin{bmatrix} 1 & 0.65 \\ -0.28 & 1 \end{bmatrix} \tag{15}$$

The joint distribution of the mixtures points out the super-Gaussian nature of the sources (see Figure 3). The matrix performance index for this simulation was $E(W, A) = 0.113$, with Crosstalk1 (E_{s1}) = -29 dB and Crosstalk2 (E_{s2}) = -34 dB. In Figure 3 it is shown how the algorithm searches for the lines of higher density instead of the contour plot.

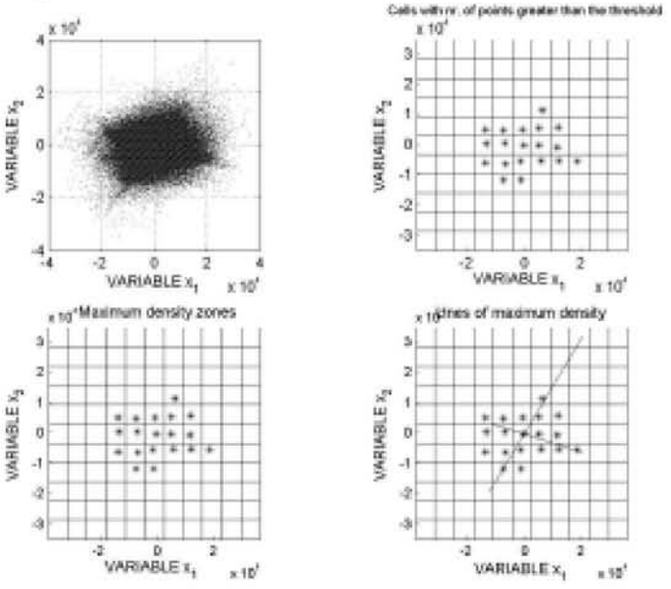


Fig. 3. Performance of the LatticeICA algorithm for a two real voice signals mixture.

3.2 Comparison with other algorithms.

In this simulation we started a more systematic exploration of the algorithm and compared the results to those obtained with two other algorithms, the FastICA [5] and Jade algorithms [4]. We tried random mixture matrixes over uniform and Laplacian mixtures of 10000 samples, running 100 simulations each time, with automatic parameters. With FastICA the number of bins has been choosen in all cases to be 180. The NRMS (normalized root mean squared error) in each case and the corresponding average convergence times (Pentium IV 1.5 GHz., 512 MB RAM, under Matlab environment) are summarized in Table 1. Although, both FastICA and Jade algorithms globally get better results than LatticeICA in most of the simulations, LatticeICA shows a great performance especially for super-Gaussian mixtures (speech signals) and it outperforms previous geometric algorithms. As a particular advantage of LatticeICA when compared with FastICA and Jade it remains its easy hardware implementation, due to the fact that it only computes simple arithmetic

operations. Future enhancements in fine tuning the radius of exclusion and adjusting the final separation lines will certainly lead to a better performance.

Table 1. Comparison of performance of the algorithm (LatticeICA) with FastICA and Jade.

Source Type	Procedure	NRMS	Speed of convergence (ms.)
Uniform	Lattice ICA	0.054	808
	FastICA	0.021	501
	Jade	0.028	584
Laplacian	Lattice ICA	0.034	703
	FastICA	0.087	406
	Jade	0.009	273

3.3 Extension to higher dimensionality.

Finally, we show how this algorithm can be extended to higher dimensionality situations by attempting to separate the projections of p mixed signals from ¡ ^{p} onto ¡ ^{2}. The signals are shown in figures 4 and 5 (with a Laplacian noise, a music source and a speech signal). The original and obtained matrices are:

$$A = \begin{bmatrix} 1 & 0.5 & 0.5 \\ 0.5 & 1 & 0.5 \\ 0.5 & 0.5 & 1 \end{bmatrix} \qquad W = \begin{bmatrix} 1 & 0.64 & 0.73 \\ 0.45 & 1 & 0.63 \\ 0.46 & 0.47 & 1 \end{bmatrix} \tag{16}$$

In Figure 4 can be seen the 3-dimensional mixture and the projections in each of the planes which will be the inputs to the algorithm. Figure 5 depicts the separated signals of the proposed LatticeICA algorithm.

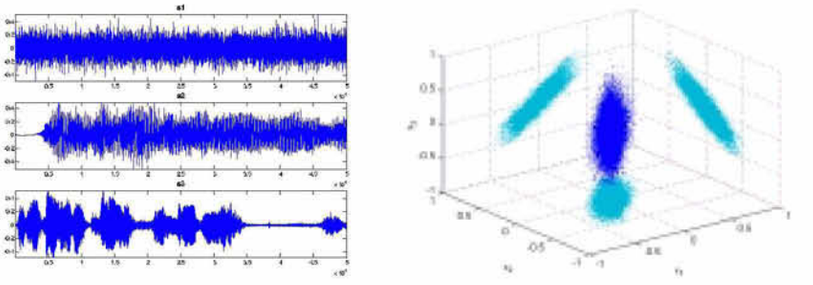


Fig. 4. Left: a laplacian noise, a music and a speech source signals. Right: three-dimensional mixture and projections.

4 Conclusions

We have developed a new geometry-based method for blind separation of sources which greatly reduces the complexity and computational load inherent in the standard

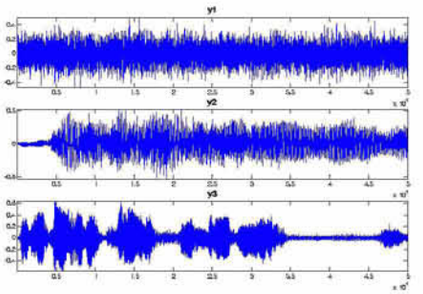


Fig. 5 Estimated signals for Simulation 3.3.

geometric ICA algorithms. This new algorithm is based on a tessellation of the input space where in each cell a code book vector is determined to represent the center of gravity of the local distribution of sample vectors. Depending on the type of distribution, either sub- or super-Gaussian, the slopes of the border lines or the diagonals are determined to obtain the coefficients of the estimated mixing matrix *W*. The method lends itself for an easy hardware implementation and is also very intuitive in terms of computer applications. Furthermore, this method could be used to detect the perimeter or outlines in simple two-dimensional figures. In the future we will intend to implement this method for more than two signals without using projections but working in the p-dimensional space.

Acknowledgement

This work has been supported by the Spanish CICYT Projects TIC2001-2845 "PROBIOCOM - Procedures for Biomedical and Communications for BSS" and TIC2000-1348 "Hybrid procedures for parallel optimization in clusters".

References

1. Álvarez, M. R., Puntonet, C. G., Rojas, I.: Separation of Sources based on the Partitioning of the Space of Observations. Lecture Notes on Computer Science, Vol. 2085. Springer-Verlag, Berlin-Heidelberg-New York (2001) 762 – 769.
2. Amari, S. I., Cichocki, A., Yang, H. H.: A new learning algorithm for blind signal separation, Proceedings of NIPS'96, (1996) 757-763.
3. Bell A. J., Sejnowski, T. J.: An information-maximisation approach to blind separation and blind deconvolution, Neural Computation, Vol. 7 (1995) 1129-1159.
4. Cardoso, J. F.: High-order contrasts for independent component analysis. Neural Computation, Vol. 11, n° 1 (1999)157-192.
5. Hyvärinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. Wiley & Sons, New York (2001).
6. Jung, A., Theis, F. J., Puntonet, C. G., Lang, E. W.: FASTGEO: A histogram based approach to linear geometric ICA. Proceedings of ICA'01 (2001) 349-354.
7. Jutten, C., Hérault, J., Comon, P., Sorouchiary, E.: Blind separation of sources, Parts I, II, III. Signal Processing, Vol. 24, n° 1 (1991) 1-29.
8. Puntonet, C. G., Prieto, A.: Neural net approach for blind separation of sources based on geometric properties. Neurocomputing, Vol. 18 (1998) 141-164.

i fr y i r r i
r f i

t l d ll

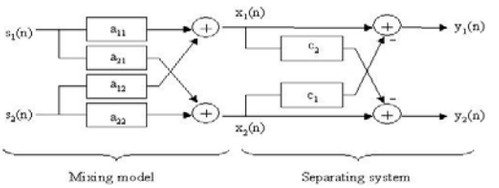
a ora o r co q é ro og n r n a on n r é a
Sa a r R Ro Nar onn 0 o o x R N
b y i t y e i e i t

A st t In a r w n ro c a n w n o rc ara on
SS o or n ar n an an o x r w c on a
o rc o ncorr a I a on g n r q nc
n n r a co r nc nc on o o r gna ara
r a o o c r q nc on w r on on
o rc ac S c on ar n o n r q r a
ra ng co ffic n an o ra o o ow r c ra n o
o r gna SS o r g r or anc or x
r o c an or no gna SNR ro n rang ro
0 o or an 0

n uc i n

l d s s t SS s d t l s l ss l SS
t ds t st s s ls s d d s ls w
st t d d s s s d d ll d s l st t s
t s t s s s ls 2 SS ld s w dl
w s t s t s t st ts l t s s lds
s t l t s s t s s l d w t t
d l w
st t SS s t t d l d s d d
d t t l s s w ss s t s s t d
st t st t st ll d d t s ls 2 w SS t ds s d
t l s s st d ls t d
t s st t s s sl d l d l ss l
SS t ds s d t d l z t t l ss SS t ds
l d s s s d t t t t s s t s d
s ls S t 7 t s s t l t
d w s t s l d sl t d ff s t
s t t s t s s
t s w s SS t d w l s t s s
t l t d d w l ts t ss d d ff s tw t s
s s t l s t t t t s t d
d d t t t s t s

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 289-296, 2003.



F on r con g ra on

p s

1 rce e r c r

t s s t s d w s d t t
s t d w t tw t d s ls d 2
d d s t tw s s s s s ls t d tw l
d d s s ls d t d d 2 d ss d t
t d d l t d s d ll d st d st t s
l t s s d s ls t ss d s

2 2
2 2 22 2

w s ts t l ffi t s t s s
s t s st s w lds t ll w t t s ls

c 2 2 2 c 22
2 2 c_2 2 22 c_2 2 2

I e c he e r y e

s d SS s t d t l t l
ts c c_2 s t t t t t s ls d 2 t s t
s st d d s t l l t s s ls s t t

2 2

2 2
d t 2 tw l s c c_2 t t s d t

c c_2 $\frac{2}{22}$ $\frac{2}{}$ o c c_2 $\frac{}{2}$ $\frac{22}{2}$ 5

t t t t l s l s ss d s

c c_2 $\frac{}{2}$ $\frac{2}{}$

w t s t l d
t d s d t s t l t d t t s l s l s
s s d t w t d ss s t l d s t s t s d s ls
s ss l st t d s t l w d ws d d t s ls
d d S s t 2 t s d s t s ss d d t

S ^{2} $S_{1\,1}$ $^{2}_{2}$ $S_{2\,2}$ o 7
S S S {

ss t t d d d t t l t l
d t l l s s t $S_{l\,l}$
d $S_{l\,l}$ 7 d s t

S ^{2} $S_{l\,l}$ o
S $S_{l\,l}$ {

s lds

$$\frac{\frac{1\ 2}{2\ 2}\ \frac{1,\ 1}{1,\ 1}}{\frac{1l}{2l}}$$

S l l t d $_{2\ 2}$ l s s t t

$$\frac{\frac{2\ 1}{1\ 1}\ \frac{2,\ 2}{2,\ 2}}{\frac{2}{1}}$$

l l s d d w t { s t d t l
t t s d l s d d
SS t d t t w s t s d tl s lts t s
t t s s s t t t s s d $_{2}$ t sts t
l st z w l t s s s t l t s d t
s s l t s s ls t d t t s t ll w st s

t d t ts tw z s d $_{2\ 2}$ w t t s s
s t l d w t t d s d t t s d s d d
S s t 2
t d t s t s t ffi ts s t d d t s
tw t w s t l d s t s t d s ls t s ts

c $\frac{\frac{1\ 2}{2\ 2}\ \frac{1,\ 1}{1,\ 1}}{}$ c_2 $\frac{\frac{2\ 1}{1\ 1}\ \frac{2,\ 2}{2,\ 2}}{}$

t d s t t t s ls s t SS s st s w

e ec he e rce e re e cy z e

t d t t w s t d t t t z s w l s
s t s s d t t t s t s st t d
t t l w d ws d d t d d s ll d l
t s d d

$\gamma_{1\ 2}$ $\frac{1\ 2\ ,}{\sqrt{1\ 1\ ,\ \cdot\ 2\ 2\ ,}}$ 2

d t ss t d l t ds $_{1\ 2}$ $\gamma_{1\ 2}$ ^{2}
t s ls d d S s t 2 d t 7 w

$$\gamma_{1\ 2} \quad \frac{^{11\cdot\ 21\cdot\ 1\ 1\ ,}\quad {}^{12\cdot\ 22\cdot\ 2\ 2\ ,}}{\sqrt{{}^{2}_{11\cdot\ 1\ 1\ ,}\quad {}^{2}_{12\cdot\ 2\ 2\ ,}\quad {}^{2}_{21\cdot\ 1\ 1\ ,}\quad {}^{2}_{22\cdot\ 2\ 2\ ,}}}$$

l t t s t s t s t ll w tw
t s d d tl z

r er y 1 *f o l o ou o* $_{1\ 2}$

r er y *f o ou o f* $S_{1\ 1}$ *d* $S_{2\ 2}$ $_{1\ 2}$

r *u d follo o o*

$$v \quad \frac{\sqrt{S_{1\ 1}}}{2\sqrt{S_{2\ 2}}} \qquad v_2 \quad \frac{^{2}\sqrt{S_{1\ 1}}}{22\sqrt{S_{2\ 2}}}$$

T o l o fu o γ *(u l o*

$$\gamma \quad \frac{v \quad v_2}{v \quad v_2}$$

odu U o (ob

γ ^{2} 5

o o γ ^{2} *f d o l f v* v_2 *T d -*
o d o o o u l o

$$v \quad v_2 \qquad {}^{22} \quad {}^{2\ 2} \sqrt{S_{1\ 1} \qquad S_{2\ 2}}$$

o o u d o b bl o ^{22} $^{2\ 2}$
T o l olu o o ob v v_2 $S_{1\ 1}$ *o* $S_{2\ 2}$
o u l o f o l o ou o d d TF o
s l s z s d $_{2\ 2}$ t d t t d t
s d t tw t st l s $_{1\ 2}$ t d t
t d t t w t ll s d t t s z s d t ss
t d s t ffi ts t s s ll ws st t t l s
t t ll t s t d w d ws t d s ls
d ll s t d d l st t s z s s
t t t s d t d s l s st z s t s
d d l st t s s l s z s s d t st z t s l st d
d t t s d l t s t ffi t *c* s
t s t t z s t d d l st s ll ws d t t
l c_2 s d t t t z s d w t
d ff tw c_2 d t s *c* s s d d t s ld
t s d t t t c_2 d s t s d t t s t s s *c*
t s d t s t t t t ffi ts d d w d t d
d t ffi t d t t d ds t w s t st s
t d t t z t d d l st t l t t t ffi t c_2
s d t t t s s d

p i n su s

st ts d s d t s s t t t l t
l st t d t l st t s t s s d d tw l s
s s ls s l t d t lt t d t s t d s l d
s s t s l w s d l t d t l t 2 s l s
s t st d t t

s t t 2 st t s t w t d ss
s t l d s t s S d S t s t s t w
d ws d d t s t d t t w s d t t s d s
d d s d d s l t w d w
t s d s ls s t s s t d s l w t
s w d ws w t 5 2 s l s t t s t w
s t l d ss s t l d s t s t s t d s ls st l l t d
s w d w d t d t s s w d ws t s ld
st t S S s l w s
w t w d ws w t 75% l tw d t s w d ws
ll w d ws s d st t t s t l d s t s d t s
d t d t t s l s z s t s l s 5 % t l
l
s d s l t s t t s t s s l
t t t d t t d t t s s w t
s d s l s t s S t s s d s ls
d t d S R_n t s t ss ss d t s t
S s d t t t SS s st d t d S R_o l w
d t S t d t s st l
S R S R_o S R_n ls t t d S
t t tw ls t SS s st S R $\sqrt{S\ R\ S\ R}$
d ts w t s l s s s ls t
s d d d t s t s s ls w t d s 2 2
s t t s d s lts l t s t sts w d t
d s t ffi ts c d c_2
w ld t d t t s ls w s t s t ffi ts
t d t t tw t sts c d c_2
s lt t d t t s ls S R t 5 t d
t s ts w t s d l t d
t lds
l t t s st t l s t w d 2 ll s
ls 75 s l s w s d t t t l t ts
s s ls s w t tw t l ts 2 w l t d
s ls s t d t ddl l ts d t s ls t t d s
t s st s w t tt l ts t t t d s t t l w

In c o c gna			
	R	*R*	*R*
		0	
0			

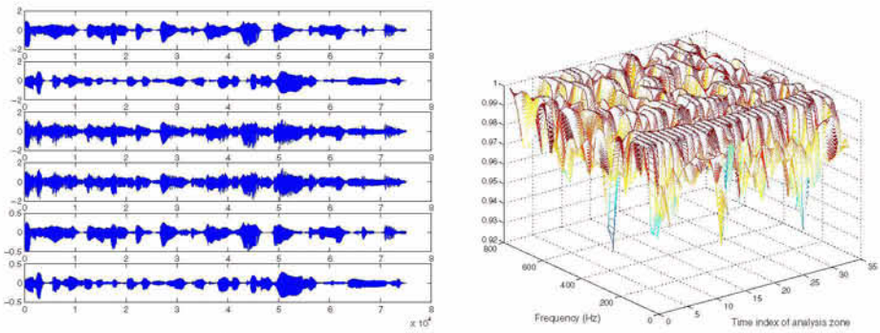
oc a ar ro SNR ro n or ar o co o c gna

d ff tw t tw s d s ls t d s ds t t t t s ls w s l t t s s ls t ls t t t s s s ls s d d t s l d t l d l t l s l s s d t s t l s t t s s s ls w t l w s st t t d s t d s s t s l s s s t t t s d d 2 s s ws t t t s t s l s t z s t t t st s l s z s w s t s w t s d t d s s s ls s s d s d t s t t s t s s ls w s w t t t st z s w l s s t s ll t s s w s ds t l s d s t l w s ts st t s t 2 z w s t l s s w l s t s s d s l s z s t s t d *f* 2 z l l t d t s t t 2

s d s d d s t ts s t s t l s ss l s t tw l s s ls s t s s s l d t l s s l s t t s ll t s s ls

In x o c gna	*R*			In x o c gna	*R*		
	a a	a a	a a		a a	a a	a a
			0				
	0					0	
	0		0				
				0		0	
	0		0 0		0		0

2 S c n anc n SNRI or ar o c gna an ca ac or



(a) (b)

Fig. 2. Source, mixed and output signals (a) and time-segmented real coherence function of observed signals (b).

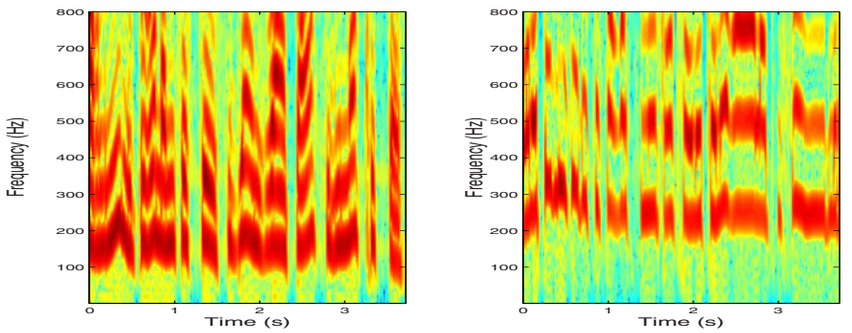
are limited to a length of 100000 samples. Moreover, a scale factor α is applied to the second source to investigate the dependence of performance with respect to the SNR available before BSS processing. In Table 2, we present $SNRI(1)$, which measures the quality of the extraction of the speech signal on channel 1. This shows that the proposed method again yields a major SNR improvement (about from 50 to 95 dB), and particularly in the most difficult condition ($\alpha = 3$) where these experiments result in an average $SNRI(1)$ equal to 78.2 dB.

4 Conclusion

In this paper, we presented a new Time-Frequency BSS method for linear instantaneous mixtures. This approach is based on the frequency-dependent real coherence function of time-segmented observations and only assumes the sources to be uncorrelated. It was tested on speech and/or noise signals and yields a very high SNR improvement, i.e. between about 50 and 95 dB. This method is here presented in the 2-channel configuration for the sake of clarity but is straightforwardly extended to any number of sources. Our future investigations will aim at extending it to convolutive mixtures.

References

1. J.F. Cardoso : Blind Signal Separation : Statistical Principles, Proceedings of the IEEE, vol.86, no.10, 1998.
2. A. Hyvarinen, J. Karhunen, E. Oja : Independent Component Analysis , John Wiley, 2001.
3. A. Belouchrani, M.G. Amin : Blind source separation using time-frequency distributions : algorithm and asymptotic performance, Proc. ICASSP'97, pp. 3469-3472, Munich, Germany, April 31-24, 1997.



a

F S c rogra o o rc gna a a c an a c

o o ar o Doncar D a a S ng a o r c on or n
o rc ara on n r q nc an roc SI 00 o o ranc
S
G r N r on or a rq n o rc ara on ng n ar
an q a ra c r q nc r r n a on roc I 00 San
D go a orn a D c 00
Jo r n S R c ar a n ara on o o n or ogona gna
x ng N o rc ro x r roc I SS 000 o
I an r J n 000
S R c ar R a an J Ro ca R a r q nc a n o rc a
ra on roc I 00 San D go a orn a D c 00
rar D n w o rc ara on a roac a on
r q nc ana or n an an o x r r roc S 00
o o ranc a 0 J n 00
rar D ro n o rc ara on o n o rc
canc a on n n r r n ca a n w a roac a on r q nc
ana roc I 00 San D go a orn a D c 00
0 D rn r n N ar n Dé arc ana c ra n n n r ré a on
a o a q a ca on à n gna ngr nag roc GR SI o
Gr no ranc
o D S g n a on an ara on o c an or no gna
ng co r nc nc on an ow r c ra roc I 00 Nara Ja an r
00

ISFET Source Separation based on linear ICA

Sergio Bermejo

Department of Electronic Engineering,
Universitat Politècnica de Catalunya (UPC),
Jordi Girona 1-3, C4 building, 08034 Barcelona, Spain
e-mail: sbermejo@eel.upc.es

Abstract. This paper addresses the separation of ion activities in ion-selective field transistor (ISFETs) arrays. These solid-state electrochemical sensors are designed to be sensitive to one kind of ion. However, their response is a non-linear function based on a mix of several ion activities found in the solution. Hence, blind source separation (BSS) techniques can be applied to recover the original main ion activity from the mixed response. Although the non-linear recovery problem could be considered difficult at first glance, our work shows how the use of prior knowledge in BSS based on a local ISFET model allows the transformation of the original problem into a linear BSS model through a simple pre-calibration step. In this context, the overall processing system is capable of recovering not only the original main ion activity but also the interfering ion activity, which has classically been treated as noise. Several preliminary experiments based on a linear ICA algorithm look promising. This work is in progress, as a part of the SEWING EU project (contract IST-2000-28084).

1 Introduction

In recent years, the use of sensor arrays has aroused a lot of attention. Unlike traditional sensor design, sensor arrays are based on the idea of using poor-selective sensors combined through a post-processing stage, which performs a kind of intelligent signal processing in order to overcome the inherent limitations of the single elements of the ensemble. Electronic noses [1] are a notorious example of the application of array techniques in chemical sensing.

In chemical sensing, analytical tasks are performed to obtain qualitative or quantitative information about particular chemical components [2]. This kind of analysis of a chemical solution can be performed with electrochemical sensors [3], which convert chemical information into an electrical signal using a transductor capable of recognizing some chemical components.

Ion-sensitive field-effect transistors (ISFETs) [4] are among the most interesting chemical sensors that have appeared in the last decades. They are built on MOSFETs [5] and achieve selectivity to a particular ion activity in a chemical solution by replacing the metallic gate in a MOSFET with a structure of membranes. Many different types of ISFETs have been presented varying the materials and structure of

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 297-304, 2003.

the sensitive membrane, e.g. MEMFETs [6], REFETs [7] and CHEMFETs [8]. However all these devices operate with the same basic principles.

In the case of ISFETS, array processing remains a necessary step not only to allow multi-component analysis but also to enhance their selectivity since ISFET response depends on the activity of a particular ion and other secondary ions, which are considered in principle as interference. Accordingly, some separation of the overall response, modeled as a (non-linear) function of a mixture of ion activities found in the solution, into single responses must be performed in order to obtain a quantitative measure (i.e. ion activities) of separated chemical components. This recovery of the original chemical information from a mixed response is denoted hereafter as ISFET source separation (ISFETSS) since it can be considered as a special case of blind source separation (BSS) [9]. However, we remove the term "blind" since we incorporate prior knowledge based on ISFET modeling into the ISFETSS model in order to characterize how the mixing is produced.

The rest of the paper is organized as follows: Section 2 presents an overall processing architecture based on an ISFET array and discusses how chemical information about single components can be recovered. In Section 3, some experiments based on artificial data generated from real ISFET measurements are introduced to demonstrate the interest of our approach. Finally, Section 4 presents some conclusions.

2. ISFET Source Separation

2.1. Prior knowledge in ISFETSS based on a local ISFET model

Since the emergence of the first ISFETs in the seventies, much theoretical work concerning ISFET modeling has been classically centered on developing an equivalent model formed by two (relatively) uncoupled parts: the electrochemical part that models the membrane potential and the electrical one based on a MOSFET model (see e.g. [8]). Until the eighties, the chemically-induced membrane which is the potential membrane was modeled through the application of the Nernst equation derived in electrochemistry (e.g. [10]). This approach can be extended by replacing the Nernst equation with the Nicolsky model [11], obtained for potentiometric glass sensors, in which interference ions are taken into account through selectivity coefficients. As noted in the eighties, these models are not accurate for describing the device globally and second-generation models were proposed (e.g.[12]). However, simple ISFET models based on the empirical Nernst and Nicolsky equations can be employed for describing their local behavior very accurately, i.e. around an operating point. According to the above approach, the steady-state drain current of an ISFET can be expressed in the linear range of operation using the typical MOSFET equations plus a threshold voltage chemically modified by the Nicolsky model for ion interference:

$$\mathrm{Id} = \beta\left(\mathrm{V_r} + \mathrm{E_o} + \Phi \log_{10}\left(\mathrm{a_i} + \sum_{\mathrm{j}} \mathrm{K_{ij}} \mathrm{a_j} \frac{\mathrm{Z_i}}{\mathrm{Z_j}} \right) - \mathrm{V_T} - 0.5\,\mathrm{Vds} \right) \mathrm{Vds} \tag{1}$$

where V_r is the voltage applied to the ISFET reference electrode, V_{ds} is its drain-source voltage, a_i is the activity of the main ion, K_{ij} is the selectivity coefficient which relates the response to the interfering ions a_j, Z_i is the valence of the main ion and Z_j is the valence of the disturbing ion j in the solution and $\{\beta, E_o, \Phi, K_{ij}, V_T\}$ are physical parameters of the model to be determined from measurements. **Fig. 1** shows the quality of a local ISFET model based on **(1)** fitted to real measurements extracted from a potassium ISFET in which a sodium interference was present [13][14]. As one can observe, this local approach to ISFET modeling is simple and accurate enough to be incorporated into ISFETSS algorithms to create a model of how the mixing is produced.

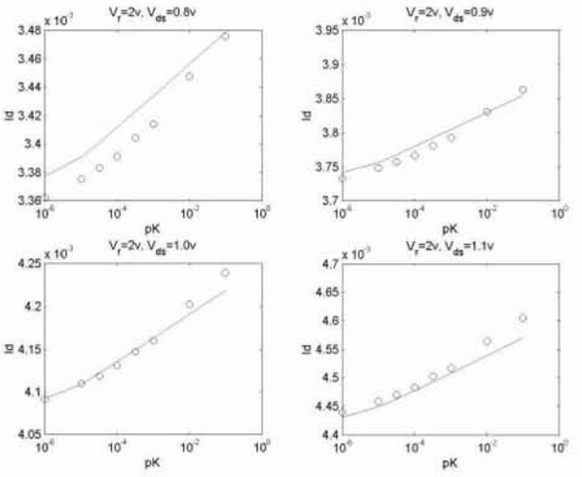


Fig. 1. Local model based on **(1)** for a Potassium CHEMFET, fitted for V_r=2 and V_{ds}=1v while it was tested for V_r=2V and V_{ds}=0.8, 0.9 and 1.1v. The estimated parameters of the model are β=.0013, E_o =.3160, K=1.93e-005, Φ=.0216, V_t =-1.4595. Note: pK=$-\log_{10}(a_K)$ and a_{Na}=0.1.

2.2. The ISFETSS Model

In the simplest ISFETSS model, we observe m discrete-time signals $x_1[n],\ldots, x_m[n]$ that correspond to a non-linear mixture of p source signals $a_1[n],\ldots, a_p[n]$ based on **(1)**, i.e.

$$\begin{aligned} \mathrm{x_1[n]} &= \mathrm{b_{11}} + \mathrm{b_{12}} \log_{10} \mathrm{s_1[n]} \\ &\vdots \\ \mathrm{x_m[n]} &= \mathrm{b_{m1}} + \mathrm{b_{m2}} \log_{10} \mathrm{s_m[n]} \end{aligned} \tag{2}$$

with

$$s_u[n] = a_i[n] + \sum_j K_{ij}^u \, a_j[n]^{\frac{Z_i}{Z_j}}, \quad u = 1,...,m \tag{3}$$

or expressed in a vector form

$$\mathbf{x}[n] = (x_1[n]...x_m[n])^T = \mathbf{F}(\mathbf{a}[n]) \tag{4}$$

where **F** is a non-linear mixing function from $\Re^p$ to $\Re^m$, T denotes transpose and the observed signals $\{x_u, u=1,\ldots,m\}$ correspond to the output voltages of m ISFET conditioning circuits whose transfer function is a linear function of the drain current in the linear range of the ISFET (see e.g. [15]). Observe that any linear function of **(1)** can be arranged in the form of **(2)**.

Then, the goal of ISFETSS is to create an inverse of **F** using a set of examples **D** extracted from **x**[n] in order to recover the latent (or hidden) variables **a**[n]. To compute a feasible solution using small training sets, some probabilistic models of the hidden variables **a**[n] should be assumed, e.g. stationary and uniform within a range of measurements.

According to the ISFETSS model in **(4)** and given the observable vector **x**, we must compute a pxm separating function **G** which allows an estimation of the source signals $\{a_i[n], i=1,\ldots,p\}$ using the following reconstruction algorithm,

$$\mathbf{y}[n] = (y_1[n]...y_p[n])^T = \mathbf{G}(\mathbf{x}[n]) \approx \mathbf{a}[n] \tag{5}$$

where clearly **G** must tend to $\mathbf{F}^{-1}$ to obtain a good estimation of **a** with **y.** In its general form **(5)** remains a difficult problem. However, as we incorporate prior knowledge about how mixing is produced into the ISFETSS model, we can perform a non-linear transform of **x** as follows

$$z_u[n] = 10^{\frac{x_u[n]-c_{u1}}{c_{u2}}} \approx s_u[n], \quad u = 1,...,m \tag{6}$$

with $\{c_{uv}\}$ are estimators of $\{b_{uv}\}$ in **(2)**. Additionally, we can write

$$\mathbf{y}[n] = \mathbf{H}(\mathbf{W}\,\mathbf{z}[n]) \tag{7}$$

where **W** estimates the inverse of the mixing matrix of ion activities based on the selectivity coefficients $\{K_{ij}^u\}$ and **H** is a non-linear function that de-emphasizes the term z_j/z_i over the reconstructed ion activities.

2.3. An ISFETSS algorithm

As one can observe from the reconstruction algorithm based on **(5)(6)(7)**, the original non-linear recovery problem in the original space remains as a particular form of linear BSS in the transformed space Z. If coefficients c_{uv} were estimated as a previous step then the ISFETSS problem could be solved using standard BSS algorithms. In our case, this estimation is possible through a supervised learning

process due to selectivity coefficients K being typically lower than 10^{-4} and consequently the drain current of an ISFET can be simplified for high main ion activities ($>10^{-2}$) as

$$\mathrm{Id} \approx \beta(\mathrm{V_r} + \mathrm{E_o} + \Phi \log_{10} \mathrm{a_i} - \mathrm{V_T} - 0.5\,\mathrm{Vds})\,\mathrm{Vds} \quad \textbf{(8)}$$

Furthermore, an additional post-calibration step using linear regression is suggested for recovering the original amplitudes of ion activities. Accordingly, we propose an integration of linear regression into a linear ICA algorithm by applying pre- and post-calibration steps as follows:

1. **Pre-calibration step**: perform linear regression to compute $\{c_{uv}\}$ using a supervised training set $\mathbf{D}_1=\{(a_{iu}, \mathbf{x}_u), u=1,\ldots,N_1\}$ where the main ion activities must belong to a range in which inferences does not affect the ISFET response. In the ideal case, a two-point pre-calibration is possible, i.e. the number of pre-calibration samples $N_1=2$.

2. **Normalization step**: remove mean m_u and scale with the root squared of sample variance σ_u each pre-calibrated component $\{z_v, v=1,\ldots,m\}$ using an unsupervised training set formed by the observed signals $\mathbf{D}_2=\{\mathbf{x}_u, u=1,\ldots,N_2\}$ extracted from measurements. Since the sample covariance matrix $\mathbf{covar}_{ZZ}$ tends to be singular, the separate normalization of each component avoids the singularity problem in the whitening process performed in several ICA algorithms.

3. **Linear ICA step**: compute **W** with a linear ICA algorithm the normalized training set formed with the pre-calibrated and normalized vectors **z'**, $\mathbf{D}_2'=\{\mathbf{z}_u, u=1,\ldots,N_2\}$.

4. **Anti-normalization step**: add m_u and multiply scale by scale factor σ_u performed in step 2.

5. **Post-calibration step**: perform linear regression to compute a relation between the original ion activities **a** and the reconstructed activities **y** using a supervised training set $\mathbf{D}_3=\{(\mathbf{a}_u, \mathbf{x}_u), u=1,\ldots,N_3\}$ where the N_3 samples of ion activities are distributed across the range of array operations. Clearly, $N_3>2$ since the linear BSS performed by the ICA algorithm is not perfect. Additionally, the independent components could be ordered to detect to which component ion activity belongs . Also, an ion activities normalization step using **H** could be necessary if $z_i \neq z_j$.

One self-evident advantage of such an approach is that the ICA algorithm does a lot of the separation work so that the linear regressors can be estimated using a very small set of supervised training samples. This feature is very interesting in chemical sensing, since multiple-point calibration is a laborious and time-consuming process. With our method, only the post-processor stage seems to require several supervised points (>2) in order to calibrate the ISFET array completely.

3. Experiments

In this section, we present some simple experiments with a 2-potassium ISFET array in which training data has been artificially created from real measurements [14]. Interferences are reduced only to the most relevant: sodium ions. Using the potassium ISFET measurements, we have extracted the ISFET model parameters of **(1)**. Then, in order to simulate the differences between the real devices of the array, we have generated random parameters uniformly distributed at ±10% around the extracted parameters. Potassium and sodium activities were uniformly distributed over $[10^{-6}, 10^{-5}]$ and [1,10] respectively for simulating a strong interference in the presence of slight potassium activity, having in mind that $K \sim 10^{-5}$. An ideal two-point pre-calibration process is assumed.

Fig. 2 shows 100 samples of potassium (main) and sodium (interferent) activities and the outputs of the pre-calibration system (step 1) which corresponds to the potassium activity plus a noise based on the sodium activity signal. The original ion activities can be almost entirely recovered applying a linear ICA algorithm like FastICA [16], as the top graphs of **Fig. 3** show. However only after applying a post-calibration step based on linear regression can we recover their original amplitudes (**Fig. 3** bottom). On the other hand, as **Fig. 3** displays, the main ion activity is reconstructed with higher accuracy than the interferent ion activity.

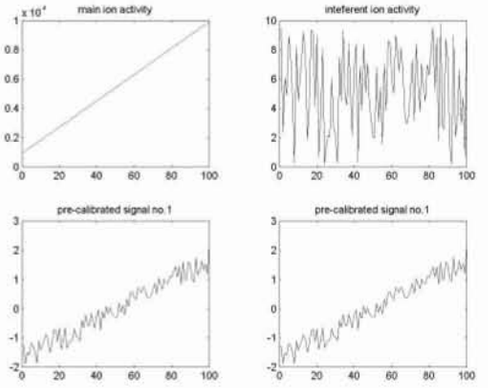


Fig. 2. Original ion activities and the outputs of the pre-calibration step.

A set of experiments, which are summarized in Fig. 4, were developed in order to analyze the quality of the reconstruction of the main ion activity depending on the number of training samples employed for the ICA and post-calibration using linear regression. Given a fixed supervised training for post-calibration (N_3=10), the relative MSE of the reconstructed potassium activity ($MSEr_K$) measured with 1000 test samples was in mean: .0052, .0029, .0009, .0007 and .0005 for N_2=100, 200, 300, 400 and 500 respectively. As expected, $MSEr_K$ monotonically decreases as the BSS performed with FastICA is better which occurs as the number of unsupervised training samples increases (N_2). It was also observed that $MSEr_{Na}$ quickly achieves a

minimum which does not improve as N_2 increases, which denotes that an optimal reconstruction point of the interference could be achieved by the ICA algorithm.

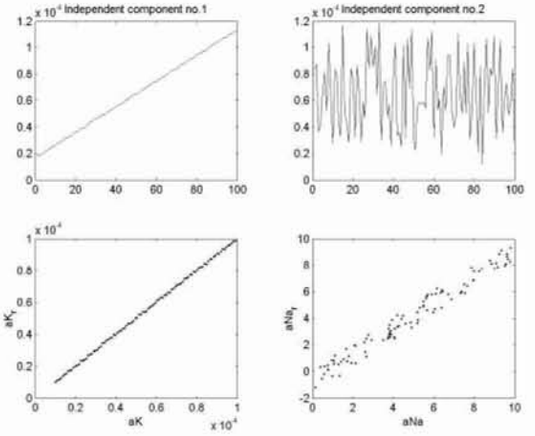


Fig. 3. Independent components signals and original vs. recovered ion activities.

For a fixed unsupervised training set (N_2=100), the evolution of $MSEr_K$ and $MSEr_{Na}$ measured in the training set is not especially affected by the increase of supervised training samples for post-calibration. However, the test $MSEr_K$ achieves a minimum for N_3=30. This behavior indicates that the performance of the system is strongly dependent on the solution achieved by the BSS stage and the increase of N_3 does not improve the reconstructed ion activities once a sufficient number of samples are provided.

In conclusion, experiments show that a faithful reconstruction of ion activities can be achieved using a very small post-calibration set (e.g. N_3=10) with the help of an unsupervised learning process performed by the ICA algorithm. The main ion activity is better reconstructed than the interference and an increase of N_2 monotonically improves this reconstruction. Unlike pre- and post-calibration steps, the ICA algorithm can be executed on-line, i.e. while the measurement system is working, so it is possible achieving this improvement in practice at no cost.

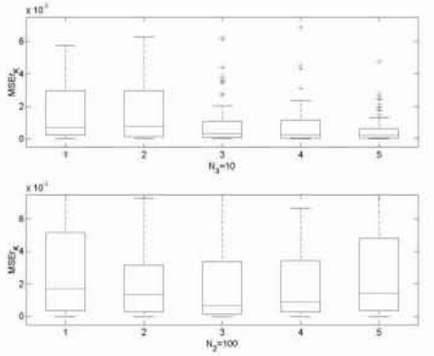


Fig. 4. Test relative MSE of the reconstructed potassium activity in 100 runs for: a) N_3=10 and N_2=100, 200, 300, 400 and 500 (top) and b) N_2=100 and N_3=10,20, 30, 40 and 50 (bottom).

4. Conclusions

An ISFET source separation (ISFETSS) model for dealing with the recovery of ion activities in ISFETs arrays has been presented. The ISFETSSS model incorporates prior knowledge extracted from ISFET models, which allows the simplification of the original recovery problem into a particular form of linear BSS. As an ISFETSS learning algorithm, we have proposed a solution that integrates linear regression into a linear ICA algorithm in such a way that a small set of supervised training samples are needed in order to calibrate the ISFET array. Experiments with artificial data generated from real measurements in a 2-ISFET array demonstrate the interest of our proposal.

Acknowledgements

We want to acknowledge Prof. Christian Jutten for his comments, which help improving the final version of this paper.

References

[1] Nagle, H. T., Schiffman, S. S. & Gutierrez-Osuna, R. (1998). The how and why of electronic noses. IEEE Spectrum, pp. 22-31.
[2] Göpel, W., Hesse, J. & Zemel, J. N. (Eds.). (1991). Sensors: A Comprehensive Survey. Volume 2. Chemical and Biochemical Sensors Part I. Edited by Göpel, W. et al., Weinheim: VCH.
[3] Wilson, D. M., Hoyt, S., Janata, J., Booksh, K. & Obando, L. (2001). Chemical Sensors for Portable, Handheld Field Instruments. IEEE Sensors Journal, Vol. 1, No. 4, pp. 256-273.
[4] Bergveld P. (1970). Development of an Ion-sensitive solid-state device for neurophysiological measurements. IEEE Trans. on Biomedical Engineering, Vol. 17, pp. 70-71.
[5] Warner Jr., R. M. & Grung, B. L. (1999). MOSFET: Theory and Design. NY: Oxford University Press.
[6] Bousse L., de Rooij N.F., and Bergveld P. (1983). Operation of Chemically Sensitive Field-Effect Sensors as a Function of the Insulator-Electrolyte Interface. IEEE Trans. Elec. Devices, Vol. 30, pp. 1263-1273.
[7] Smith R.L., and Scott D.C. (1986). An integrated sensor for electrochemical measurements. IEEE Trans. on Biomedical Engineering, Vol. 33, pp. 83-90.
[8] Sibbald A. (1983). Chemical-sensitive field-effect transistors. IEE Proceedings, Vol. 130, Pt. I, No. 5, pp. 233-244.
[9] Cichocki, A. & Amari, S. (2002). Adaptive Blind Signal and Image Processing. NY: Wiley.
[10] Moss, S. D., Johnson, C. C. & Janata, J. (1978). Hydrogen, calcium, and potassium ion-sensitive FET transducers: a preliminary report. IEEE Trans. on Biomedical Engineering, Vol. 25, No.1, p.49-53.
[11] Nicolsky, B.P. et al., (1967). in Glass electrodes for hydrogen and other cautions, Eisenman, G., (ed.): NY: M. Dekker.
[12] Barabash, P. R., Cobbold, R. S. C., Wlodarski, W. B. (1987). Analysis of the Thheshold Voltage and its Temperature Dependence in Electrolyte-Insulator-Semiconductor Field-Effect Transistor (EISFET), IEEE Trans. on Electron Devices, Vol. 34, pp. 1271-1282.
[13] Brzózka, Z., Holterman, A. J., Honing, W.N., Verkerk, U.H., can den Vlekkert, H. H., Engbersen, J. F. J., Reinholut, D. N. (1994). Enhanced performance of potassium CHEMFET by optimization of a polysiloxane mebrane. Sensor & Actuators B, Vol. 18-19, pp. 38-41.
[14] Opalski, L. J. Personal communication.
[15] Palán, B., Santos, F. V., Karam, J. M., Courtois, B., Husák, M. (1999). New ISFET sensor interface circuit for biomedical applications. Sensor & Actuators B, Vol. 57, pp. 36-42.
[16] Hyvärinen, A. & Oja, E. (1999). Fast and robust fixed-point algorithms for independent component analysis. IEEE trans. Neural Networks, Vol. 10, pp.626-634.

An application of ICA to blind DS-CDMA detection: a joint optimization criterion

Iván Durán and Sergio Cruces*

Area de Teoría de la Señal y Comunicaciones, Escuela de Ingenieros,
Universidad de Sevilla, Camino de los descubrimientos s/n,
41092-Sevilla, Spain
{iduran, sergio}@us.es

Abstract. This paper studies the application of the theory and algorithms of blind signal extraction to solve the problem of the detection of the desired users in a DS-CDMA communications system. Typically, the uplink in this system is characterized by users that transmit asynchronously and propagation channels that are multipath. We address inverse filter criterion introduced by Tugnait in [1] and we show that a prewhitening preprocessing approach together with the joint combination of several higher order statistics improves the detector performance.

1 Introduction

Direct sequence code division multiple access (DS-CDMA) is a technique widely extended in mobile communications [3–5, 1]. In these systems users share the same band of frequencies and the same time slots, but they are separated in code. Each user transmits with a a different cyclic and quasi-orthogonal code (or sequence) that multiplies the user's symbols.

The main sources of errors at the detector are due to the multi-access interference, the inter-symbol interference, the asynchronism of users and the near-far problem. The multi-access interference (MAI) is a consequence of the fact that users transmit at the same time and in the same band of frequencies. The inter-symbol interference (ISI) is due to the multi-path propagation in the channel and becomes important in systems with a high bit-rate since in this case the delays due to multi-path can spread up to the duration of one symbol [1]. The asynchronism of the users consist in that they start to transmit at a different time instants, while near-far problem is caused by the difference between the power transmission for the different users. These two last problems only occur at the up-link (from the mobile stations to the base stations), because in the down-link the base station transmits in a synchronous way and with the same power to all users [3].

Blind multiuser detection can be performed with the only knowledge of the desired users codes, avoiding the need for training sequences that lower the spectral efficiency [4].

* This work has been supported by the CICYT project of the Government of Spain (Grant TIC2001-0751-C04-04).

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 305-312, 2003.

In this work we pay special attention to the algorithm derived in [1] from an inverse filter criterion which maximizes the normalized fourth-order cumulant of the output. In this paper we show how the incorporation of prewhitening and the extension of the criterion to consider the joint optimization of several cumulant orders lead to a considerable improvement in the MSE of the detected user.

The paper is structured as follows. Section 2 presents the signal model while Sections 3 and 4.1 review the code constrained criterion and extraction algorithm introduced by [1]. In Section 4.2 we present our main contribution, the incorporation of prewhitening and the extension of the criterion to consider the joint optimization of several cumulant orders. In Section 5 we corroborate the theoretical behavior of the algorithms with simulations.

2 System Model

In the Blind Signal Extraction problem one typically considers the existence of N independent source signals $\mathbf{s}(k) = [s_1(k), \ldots, s_N(k)]^{\mathrm{T}}$. These signals are combined, in presence of additive noise $\mathbf{n}(k)$, by a linear memoryless system characterized by the matrix $\mathbf{A}$, resulting the vector of M observations

$$\mathbf{x}(k) = \mathbf{A}\mathbf{s}(k) + \mathbf{n}(k) \; . \tag{1}$$

The recovery of the desired sources can be decomposed in two steps. The first step pre-whitens the observations, whereas the second step extracts the desired source. The prewhitened observations are

$$\mathbf{z}(k) = \mathbf{W}\mathbf{x}(k) \tag{2}$$

where $\mathbf{W}$ is the matrix which enforces the prewhitening $E[\mathbf{z}(k)\mathbf{z}(k)^{\mathrm{H}}] = \mathbf{I}_M$.

To obtain the desired source we multiply $\mathbf{z}(k)$ by a separation or extraction matrix $\mathbf{U}$ obtaining the output signal or estimated source

$$y(k) = \mathbf{U}\mathbf{z}(k) = \mathbf{G}\mathbf{s}(k) + \mathbf{U}\mathbf{W}\mathbf{n}(k) \; . \tag{3}$$

where $\mathbf{G} := \mathbf{UWA}$ is the global transfer matrix from the sources to the output.

Next we will use the model of [1] to convert the detection of an user in a DS-CDMA system into a problem of BSE with a linear and instantaneous mixture.

We will study a short-code DS-CDMA system (i.e., the spreading sequence has exactly one symbol of duration). In future high-capacity systems it will become more useful than long-code because it allows the multiuser receiver to know adaptively the MAI (multi-access interference) structure since the MAI in a symbol has identical statistics that the MAI in the next symbol [5].

We consider a system with N_u users and N_c chips per symbol. The jth user's chip sequence is $\mathbf{c}_j = [c_j(0) \; \cdots \; c_j(N_c - 1)]^{\mathrm{T}}$. The symbol sequence transmitted by the jth user is $\{b_j(k)\}$. The sequence $b_j(k)$ is zero-mean, independently and identically distributed (i.i.d.). For different js $\{b_j(k)\}$s are mutually independent. They are complex because the modulation can be quadrature or binary.

The signal transmitted by the jth user is therefore

$$\hat{x}_j(n) = \sum_{k=-\infty}^{\infty} b_j(k)c_j(n - kN_c), \quad j = 1, 2, \ldots, N_u \; . \tag{4}$$

This signal will pass through the corresponding linear and dispersive channel whose impulse response sampled at the chip interval T_c is (for the jth user) $g_j(n)$, which includes the effects of chip matched filtering at the receiver (see [5]) but not the transmission delay (mod N_c) of jth user, d_j, $(0 \leq d_j \leq N_c - 1)$. If we group the effect of the chip sequence and the channel in $h_j(n)$ we can write the contribution due to the jth user at the receiver after to be sampled at T_c as

$$\widetilde{x}_j(n) = \sum_{l=-\infty}^{\infty} b_j(l)h_j(n - d_j - lN_c) \tag{5}$$

$$h_j(n) := \sum_{m=0}^{N_c-1} c_j(m)g_j(n - m) \; . \tag{6}$$

The total received signal $\widetilde{x}(n)$ is the superposition of contributions of the N_u users plus an additive white Gaussian noise $w(n)$.

Now we will define the convolutional MIMO model (multiple inputs multiple outputs) from which we will build the instantaneous MIMO model to which apply the algorithms of blind extraction. If we define $\widetilde{\mathbf{x}}(k) = [\widetilde{x}(kN_c + N_c - 1) \; \cdots \; \widetilde{x}(kN_c)]^{\mathrm{T}}$, $\mathbf{h}_j(l) = [h_j(lN_c - d_j + N_c - 1) \; \cdots \; h_j(lN_c - d_j)]^{\mathrm{T}}$ and $\widetilde{\mathbf{n}}(k) = [w(kN_c + N_c - 1) \; \cdots \; w(kN_c)]^{\mathrm{T}}$, the convolutional MIMO model is

$$\widetilde{\mathbf{x}}(k) = \sum_{j=1}^{N_u} \sum_{l=0}^{L_j} \mathbf{h}_j(l)b_j(k - l) + \widetilde{\mathbf{n}}(k) \; . \tag{7}$$

If we assume that multipaths delays are at most one symbol duration ($g_j(l) = 0$ for $l < 0$ and $l > N_c$) and recalling that $0 \leq d_j < N_c$ we have $\mathbf{h}_j(l) = 0$ for $l < 0$ and $l \geq 3$. Thus $L_j = 2$ for $j = 1, \ldots, N_u$.

To convert the convolutional MIMO model in an instantaneous MIMO model we define $\mathbf{x}(k) := [\widetilde{\mathbf{x}}(k), \ldots, \widetilde{\mathbf{x}}(k - L_e + 1)]^{\mathrm{T}}$ and $\mathbf{n}(k)$ in the same way, so that

$$\mathbf{x}(k) = \mathbf{A}\mathbf{s}(k) + \mathbf{n}(k) \tag{8}$$

being $\mathbf{s}(k) = [\widetilde{\mathbf{s}}(k), \ldots, \widetilde{\mathbf{s}}(k - L_e + 1)]^{\mathrm{T}}$ the independent sources vector and

$$\mathbf{A} = \begin{bmatrix} \mathbf{H}(0) & \mathbf{H}(1) & \mathbf{H}(2) & 0 & 0 & & \cdots & 0 \\ 0 & \mathbf{H}(0) & \mathbf{H}(1) & \mathbf{H}(2) & 0 & \cdots & & 0 \\ \vdots & & & & \ddots & & & \vdots \\ 0 & & \cdots & & & \mathbf{H}(0) & \mathbf{H}(1) & \mathbf{H}(2) \end{bmatrix} \tag{9}$$

the matrix of linear and instantaneous mixture, being $\mathbf{H}(l) = [\mathbf{h}_1(l), \ldots, \mathbf{h}_{N_u}(l)]^{\mathrm{T}}$ and $\widetilde{\mathbf{s}}(k) = [b_1(k), \ldots, b_{N_u}(k)]^{\mathrm{T}}$.

The vector $\mathbf{x}(k)$ is $N_cL_e \times 1$, the matrix $\mathbf{A}$ is $N_cL_e \times N_u(L_e + 2)$, and $\mathbf{s}$ is $N_u(L_e + 2) \times 1$. Following the typical notation of BSS and BSE, $M = N_cL_e$ (number of observations) and $N = N_u(L_e + 2)$ (number of independent sources).

3 Code constrained criterion

We will address the code constrained criterion introduced by [1] to assure the extraction of the desired user. We will assume the use of the prewhitening process and will use an equivalent expression to obtain the projection matrix needed.

Let us assume we want to obtain the symbol sequence of the user j_0 with a delay d. Then, after prewhitening the observations given by (8), the extraction matrix $\mathbf{U}$ will be a row vector that in the absence of noise must satisfy

$$\mathbf{UWAA}^{\mathrm{H}} = [0 \dots \alpha \dots 0]\, \mathbf{A}^{\mathrm{H}} = \alpha \left[\mathbf{h}_{j_0}^{\mathrm{H}}(d) \dots \mathbf{h}_{j_0}^{\mathrm{H}}(0) 0 \dots 0\right]. \tag{10}$$

where α is a complex constant that goes at the position $j_0 + N_u d$. Defining $\mathbf{h}_j^{(d)} := \left[\mathbf{h}_j^{\mathrm{H}}(d) \dots \mathbf{h}_j^{\mathrm{H}}(1)\mathbf{h}_j^{\mathrm{H}}(0)\right]^H$, and recalling that $g_j(l)$ is 0 for $l > N_c$ and for $l < 0$, and $0 \le d_j < N_c$, we have

$$\mathbf{h}_j^{(d)} = \mathbf{C}_j^{(d)} \mathbf{g}_j \tag{11}$$

being

$$\mathbf{C}_j^{(d)} := \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ c_j(N_c-1) & 0 & \cdots & 0 \\ c_j(N_c-2) & c_j(N_c-1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ c_j(0) & c_j(1) & \cdots & 0 \\ 0 & c_j(0) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & c_j(N_c-1) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & c_j(0) \end{bmatrix} \tag{12}$$

$$\mathbf{g}_j := \left[g_j(2N_c - 1 - d_j) \ \cdots \ g_j(-d_j+1)\ g_j(-d_j)\right]^{\mathrm{T}}. \tag{13}$$

$\mathbf{C}_j^{(d)}$ is a Toeplitz $[(d+1)N_c] \times [2N_c]$ matrix.

Now, using (10) and (11) and knowing that $\mathbf{AA}^{\mathrm{H}} = \mathbf{R}_{xx} = E\{\mathbf{x}(k)\mathbf{x}^{\mathrm{H}}(k)\}$ in the absence of noise, we can write

$$\mathbf{R}_{xx}\mathbf{W}^{\mathrm{H}}\mathbf{U}^{\mathrm{H}} = \alpha \mathcal{C}_{j_0}^{(d)} \mathbf{g}_{j_0} \tag{14}$$

where

$$\mathcal{C}_{j_0}^{(d)} = \begin{bmatrix} \mathbf{C}_{j_0}^{(d)} \\ \mathbf{0} \end{bmatrix} \tag{15}$$

is a $N_c L_e \times 2N_c$ matrix and $\mathbf{W}$ is the prewhitening matrix ($\mathbf{WR}_{xx}\mathbf{W}^{\mathrm{H}} = \mathbf{I}_{N_c L_e}$).

Thus

$$\mathbf{U}^{\mathrm{H}} = \alpha \mathbf{W} \mathcal{C}_{j_0}^{(d)} \mathbf{g}_{j_0} = \mathcal{C}_{j_0}^{1(d)} \alpha \mathbf{g}_{j_0} \tag{16}$$

where $\mathcal{C}_{j_0}^{1(d)} := \mathbf{W}\mathcal{C}_{j_0}^{(d)}$.

The projection matrix into the subspace of $\mathcal{C}_{j_0}^{1(d)}$ is

$$\mathbf{\Pi}_c = \mathcal{C}_{j_0}^{1(d)} \left(\mathcal{C}_{j_0}^{1(d)\mathrm{H}} \mathcal{C}_{j_0}^{1(d)} \right)^{-1} \mathcal{C}_{j_0}^{1(d)\mathrm{H}} . \tag{17}$$

We have to post-multiply $\mathbf{U}$ by $\mathbf{\Pi}_c$ in the algorithms of separation or extraction to ensure that we obtain the desired user.

If we do not use prewhitening (like [1]) then $\mathcal{C}_{j_0}^{1(d)} := \mathbf{R}_{xx}^{-1}\mathcal{C}_{j_0}^{(d)}$.

4 Extraction algorithms

In this section we begin presenting the existing implementation of the inverse filter criterion and we later consider the improvement of this algorithm with the joint optimization criterion.

4.1 Algorithm derived from the Inverse Filter Criterion

In [1] Tugnait *et al.* propose to use an inverse filter (equalizer) that apply on $\widetilde{\mathbf{x}}(k)$ to solve the extraction problem. This equalizer is adapted using a gradient algorithm that maximizes the criterion

$$J_{42}(\mathbf{B}) = \frac{|cum_4\left(y\left(k\right)\right)|}{\left(E\{|y(k)|^2\}\right)^2} \tag{18}$$

which is based on the fourth-order cumulant $cum_4(y(k)) := E\{|y(k)|^4\} - 2[E\{|y(k)|^2\}]^2 - |E\{y(k)^2\}|^2$ of the output

$$y(k) = \mathbf{B}\mathbf{x} = \sum_{i=0}^{L_e-1} \widetilde{\mathbf{B}}(i)\widetilde{\mathbf{x}}(k-i) \tag{19}$$

where $\widetilde{\mathbf{B}}(i)$ is the equalizer of length L_e symbols and $N \times 1$-dimensional.

In absence of noise, the recovery at the output of the desired j_0th user up to a complex constant $\alpha \neq 0$, and an arbitrary delay $0 \leq d \leq L_e - 1 + L_{j_0}$

$$y(k) = \alpha b_{j_0}(k-d) \tag{20}$$

is blindly achieved through the maximization of (18) with respect to the row vector

$$\mathbf{B} = \left[\widetilde{\mathbf{B}}(0), \widetilde{\mathbf{B}}(1), \cdots, \widetilde{\mathbf{B}}(L_e-1)\right] . \tag{21}$$

4.2 Joint optimization criterion

Assuming the prewhitening of the observations we have that $\mathbf{U}$ and $\mathbf{G}$ are row vectors both of unit 2-norm and the output takes the form $y(k) = \mathbf{U}\mathbf{z}(k)$.

We propose to estimate the desired independent component by jointly maximizing a weighted square sum of a combination of cumulants of the outputs with orders $r \geq 2$. A contrast function that achieves this objective is given by

$$\psi_{\Omega}(y) = \sum_{r \in \Omega} \alpha_r \left| cum_r \left(y(k) \right) \right|^2 \qquad \text{subject to } \|\mathbf{U}\|_2 = 1. \tag{22}$$

where α_r are positive weighting terms. Let us define $q = \max\{r \in \Omega\}$, in our case, we choose to optimize the set of cumulant orders $\Omega = \{4, 6\}$.

The problem with this approach is in the difficulty of the optimization of (22), which is highly non-linear with respect to $\mathbf{U}$. We can circumvent this difficulty by proposing a similar contrast function to (22) but whose dependence with respect to each of the extracting system candidates is quadratic, and thus, much more easy to optimize using algebraic methods.

Theorem 1. *Considering a set of q candidates for the extracting system $\{\mathbf{U}^{[1]}, \ldots, \mathbf{U}^{[q]}\}$ each of unit 2-norm, and the set of their respective outputs $\bar{y} = \{y^{[1]}, \ldots, y^{[q]}\}$, the following multivariate function*

$$\psi_{\Omega}(\bar{y}) = \sum_{r \in \Omega} \frac{\alpha_r}{\binom{q}{r}} \sum_{\sigma \in \Gamma_r^q} \left| cum(y^{[\sigma_1]}(k), \cdots, y^{[\sigma_r]}(k)) \right|^2 \tag{23}$$

where $\alpha_r > 0$ and Γ_r^q is the set of all the possible combinations $(\sigma_1, \cdots, \sigma_r)$ of the indexes $\{1, \ldots, q\}$ taken r by r, is maximized at the extraction of one of the users, i.e., at this extreme point $y^{[1]}(k) = \cdots = y^{[q]}(k) = b_{j_0}(k - d)$.

A sketch for the proof of this theorem is presented in [2]. There is an interpretation of this theorem in terms of a low rank approximation of a set of cumulant tensors[6]. The contrast can be easily maximized optimizing $\psi_{\Omega}(\bar{y})$ cyclically with respect to each one of the elements $\mathbf{U}^{[m]},\ m = 1, \ldots, q$, while keeping fixed the others. In the following, it will be useful to refer to the variables that take a given value at the i-th iteration, this sequential notation will be denoted with the super-index $()^{(i)}$. Due to the invariant property of $\psi_{\Omega}(\bar{y})$ with respect to permutations in its arguments, the cyclic maximization of the contrast with respect to $\mathbf{U}^{[m]}$ with $m = (i \bmod \mathrm{q}) + 1$ is equivalent to the sequential maximization of the function

$$\begin{aligned} \phi_{\Omega}(\mathbf{U}^{(i)}) &= \sum_{r \in \Omega} \alpha_r |cum_r(y^{(i)}(k), y^{(i-1)}(k), \ldots, y^{(i-r+1)}(k))|^2 \\ &= \mathbf{U}^{(i)} \mathbf{M}^{(i-1)} \mathbf{U}^{(i)H} \end{aligned} \tag{24}$$

w.r.t. the extraction system $\mathbf{U}^{(i)}$ through the iterations. Note that $\mathbf{M}^{(i-1)}$ is a constant matrix (as long as $\mathbf{U}^{(i-1)}, \cdots, \mathbf{U}^{(i-q+1)}$ are kept fixed) given by

$$\mathbf{M}^{(i-1)} = \sum_{r \in \Omega} \alpha_r \mathbf{c}_{\mathbf{z}y}^{(i-1)}(r) \ \left(\mathbf{c}_{\mathbf{z}y}^{(i-1)}(r) \right)^H \tag{25}$$

where $\mathbf{c}_{\mathbf{z}y}^{(i-1)}(r)$ may be defined as in (b) or as in (a)

$$\mathbf{c}_{\mathbf{z}y_i}^{(k-1)}(r) = \begin{cases} \text{a)} & Cum(\mathbf{z}(k), y^{(i-1)}(k), \cdots, y^{(i-q+1)}(k)) \\ \text{b)} & Cum(\mathbf{z}(k), y^{(i-1)}(k), \cdots, y^{(i-1)}(k)) \end{cases} \tag{26}$$

depending whether, after each iteration, a projection step onto the symmetric subspace $y^{[1]}(k) = \cdots = y^{[q]}(k)$ that contains the solutions is performed or not.

At each iteration the maximization $\phi_\Omega(\mathbf{U}^{(i)})$ is obtained finding the eigenvector associated to the dominant eigenvalue of $\mathbf{M}^{(i-1)}$. Starting from the previous solution, if one considers to use L iterations of the power method to approximate the dominant eigenvector (in practice $L = 1$ or 2 works well), the following extraction algorithm is obtained

$$\begin{aligned} &\underline{\mathbf{U}}^{(0)} = \mathbf{U}^{(i-1)} \\ &\text{FOR} \quad l = 1 : L \\ &\quad \underline{\mathbf{U}}^{(l)} = \frac{\sum_{r\in\Omega} \alpha_r d_y^{(l-1)}(r) \left(\mathbf{c}_{\mathbf{z}y}^{(i-1)}(r)\right)^H}{\left\| \sum_{r\in\Omega} \alpha_r d_y^{(l-1)}(r) \left(\mathbf{c}_{\mathbf{z}y}^{(i-1)}(r)\right)^H \right\|_2} \\ &\text{END} \\ &\mathbf{U}^{(i)} = \underline{\mathbf{U}}^{(L)} \end{aligned} \tag{27}$$

where $d_y^{(l-1)}(r) = \underline{\mathbf{U}}^{(l-1)} \mathbf{c}_{zy}^{(i-1)}(r)$.

5 Simulations

In this section the comparison of the algorithms is done in terms of the mean square error criterion between the output and the symbol sequence of the desired user.

We made two different simulations: one in which all users are received with the same power and one in which the difference between the power of the desired user and one of the other users is 10dB (near-far situation).

In each simulation we consider an observations time of 200 symbols and three 4-QPSK users with 8 chip/symbol. The channel was formed by four random multi-paths and the equalizer length was set to $L_e = 3$.

We compared the results of the algorithm of [1] with and without prewhitening, and the algorithm of combined cumulants with $q = 6$ and $\alpha_4 = \alpha_6 = 0.5$.

In Fig. 1 we show the results of the simulations. We can see that the prewhitening reduces the MSE between the output and the desired user and that the reduction is higher if we use the proposed algorithm.

We should note that occasionally, for very low SNR, initialization process may fail when prewhitening is considered. When this happens the algorithm fail to obtain the desired user. This could be avoided if the system is conveniently reinitialized whenever this problem is detected.

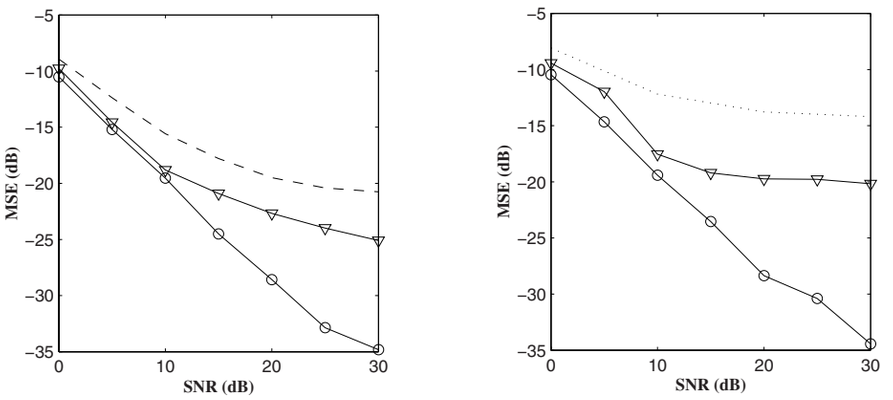


Fig. 1. Figures show the MSE between the output and the desired user over 100 Monte Carlo runs: '△' and dash line denote the algorithm (18) with and without prewhitening, respectively, whereas 'o' denotes the proposed algorithm. The MAI was 0dB for the simulation in left figure and 10dB for that shown in the right figure.

6 Conclusions

In this paper we have addressed the problem of the blind extraction of a desired user in a DS-CDMA communications system. We show how to extend the inverse filter criteria with code constrained presented in [1] to consider a more general criterion based on joint optimization of several higher order statistics. The simulations corroborate how this later criterion improves MSE between the extracted symbol sequence and that of the desired user.

References

1. Tugnait, Jitendra K. and Li, Tongtong: Blind Detection of Asynchronous CDMA Signals in Multipath Channels Using Code-Constrained Inverse Filter Criterion. IEEE Trans. Signal Processing, Vol. 49, No. 7, pp. 1300-1309, Jul. 2001
2. Cruces S., A. Cichocki, A.: Combining Blind Source Extraction with Joint Approximate Diagonalization: Thin Algorithms for ICA, Accepted in the 4rd international conference on Independent Component Analysis and Blind Signal Separation. Nara, Japan. April, 2003.
3. Schniter, Phil and Johnson Jr., C. Richard: Minimum-Entropy Blind Acquisition/Equalization for Uplink DS-CDMA. Allerton Conference on Communication, Control and Computing, 1998.
4. Gesbert, D., Sorelius, J. and Paulraj, A.: Blind Multiuser MMSE Detection of CDMA Signals, Proc. of ICASSP, 1998.
5. Madhow, Upamanyu: Blind Adaptive Interference Suppression for Direct-Sequence CDMA. Proceedings of the IEEE, Vol. 86, No. 10, pp. 2049-2069, Oct. 1998
6. De Lathauwer, L., Comon, P., De-Moor, B., and Vandewalle, J.: Higher-order power method - application in independent component analysis. International Symposium on Nonlinear Theory and Applications NOLTA, Dec. 1995.

A Genetic Algorithm for Controlling Elevator Group Systems

P. Cortes, J. Larrañeta, and L.Onieva

Ingeniería Organización. Escuela Superior Ingenieros. Seville University, Camino de los Descubrimientos s/n. Sevilla 41092. Spain
pca@esi.us.es

Abstract. The efficient performance of elevator group system controllers becomes a first order necessity when the buildings have a high utilisation ratio of the elevators, such as in professional buildings. We present a genetic algorithm that is compared with traditional controller algorithms in industry applications. An ARENA simulation scenario is created during heavy lunchpeak traffic conditions. The results allow us to affirm that our genetic algorithm reaches a better performance attending to the system waiting times than THV algorithm.

1 Introduction

The installation of synchronized elevator groups in professional use buildings (offices, hospitals or hotels) is an usual practice. The large utilisation ratio of the elevators makes necessary the implementation of such systems in order to give quality to the users-passengers and energetic efficiency to the building managers.

In this situation, it is usual to select the system waiting time as the goal to attain an efficient system performance. Also, the maximum waiting time has to be had in account as an additional limitation. The system waiting time includes the waiting time for the lift in the hall plus the trip time inside the lift.

When dealing with elevator systems is a usual practice to consider the following assumptions (where most of them are evident assumptions). Each hall call is attended by only one cabin. The maximum number of passengers being transported in the cabin is bounded by its capacity. The lifts can stop at a floor only if it exists a hall call or a cabin call in that floor. The cabin calls are sequentially served in accordance with the lift trip direction. A lift carrying passengers cannot change the trip direction.

The traditional type of controllers implemented in the industry follows simple dispatch rules that make use of an IF-ELSE logical commands set. Among these dispatch rules, the THV is one of the most habitual algorithms. The THV assigns the hall call to the nearest lift in the adequate trip direction [1].

Recently, advanced methods have been proposed showing better performance. Examples of them are the *Optimal Routing algorithm*, the *Dynamically Adaptive Call Allocation* (DACA) and the *Adaptive Call Allocation* (ACA) [2] that are based on Dynamic Programming. Also Fuzzy Logic has been proved as a valuable alternative when evaluating a large amount of criteria in a flexible manner. The fuzzy elevator

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 313-320, 2003.

group control system [3] and the Fuzzy Elevator Group Controller with Linear Context Adaptation [4] are some examples. Bio-inspired systems [5] and [6] have been revealed successful capacities. Here we propose a genetic algorithm based on a hall call allocation strategy to identify the chromosomes of the population individuals.

The rest of the paper deals with the simulation model in section 2, the genetic algorithm characteristics in section 3, the main results of the simulations showed in section 4 and the conclusions in the final section.

2 Simulation Model

We have made use of the ARENA v.5.0 software to simulate the possible event set. ARENA is a powerful interactive visual modelling system that makes use of the SIMAN language. The initial model consists of an animation zone and a module logical zone that can be divided into one controller zone, one passenger zone and two elevator zone for each of the cabins.

The ***animation zone*** is defined by the `Arrive` and `Depart` modules, which regulate the arrivals and departures of the passengers at the system.

In the ***controller zone***, one entity is created by lift to travel around the logical zone. When the passengers come into the lift, the passengers are joined to the `controller` entity shaping one only entity at the same time as holding all the particular individual entities attributes.

The ***passenger zone*** consists of the allocation of the `UpDown` attribute (1 if the passenger goes up and 2 otherwise) that is stated as function of the `Origin` and `Destination` attributes. So, the passenger is sent to the waiting queue if it exists. Otherwise the hall call allocation procedure is done by means of the correspondent optimisation algorithm (our genetic algorithm by the case).

For each one of the ***elevator zones***, when the lift arrives at a floor the subsequent actions must be checked and done if necessary: lift waits for calls, passengers leaves the lift, passengers come into the lift, lift allocation in case of full capacity, cabin call allocation and call evaluation.

When the lift arrives at a floor, the `state` of the lift is evaluated. If the lift `state` is set to zero, the lift is stopped and will have access to the `Waiting_for_Calls` submodule. If the lift is not stopped and it is carrying passengers, it inputs into the `Leaving_the_Lift` submodule. If it is not carrying passengers, it inputs into the `Taking_Passengers` submodule after a `Delay` to simulate the opening doors time (we use the delay variable `Time_Doors` (2.5 seconds).

When the lift arrives at a floor, the `Arrival_Evaluation` submodule presents three options: the lift continues up, the lift continues down or the lifts starts the deceleration process (preparing to stop). We use the `LDX (Transporter ID, unit number)` as an ARENA proprietary variable allowing to know the floor in which the lift is. We load this data in the variable `Level`.

After updating `Level`, if the lift is going up and the lift is in the ground floor, the lift is sent to the first floor; if the lift is going down and the lift is in the highest floor, the lift is sent to the last but one floor. Otherwise the simulation model checks if the lift has stopped in the floor that `Level` indicates (this data can be checked by means

of the variable `Last_Visited_Floor`), in this case the lift is sent up or down depending on the trip direction. Otherwise the lift stops at the floor if there exists a cabin call or a hall call and the capacity is not full. If the lift is full capacity and it does not exist cabin calls, the lift is sent up or down depending on the trip direction.

3 Genetic Algorithm for the Controller

We propose a genetic algorithm that makes use of a hall call allocation strategy to perform the elevator group controller. For each time, *t*, the hall calls and the cabin calls of the system are evaluated, allocating the hall calls to one specific lift. Each time, *t*, the set of hall calls are reallocated allowing the subsequent modification if the system performance improves. Each time the set of decisions is taken managing all the available information (planning for the long term) but only carrying the immediate action out for each lift of the group: stop, upwards or downwards displacement.

So, each time, *t*, the simulation model makes a call to the controller optimisation module (the genetic algorithm) that returns the overall call allocation. The genetic algorithm is defined by the following characteristics.

3.1 Individuals and Population

Two arrays of size [2·`Number_of_Floors`-2] define the individual chromosome. Each of the arrays defines the system `state` for each one of the lifts. The array is divided into two parts; the first refers to the up traffic and the second one to the down traffic.

The first `Number_of_Floors`-1 integers correspond to the hall calls in the upward direction from the ground floor to the highest floor. The second `Number_of_Floors`-1 integers correspond to the hall calls in the downward direction from the highest floor to the ground floor. Figure 1 depicts the chromosome individuals:

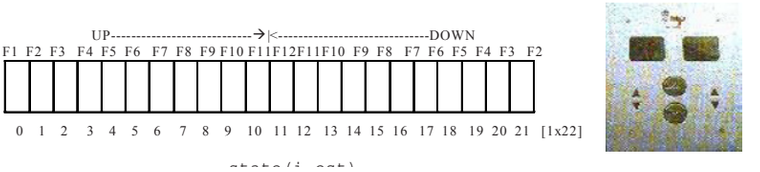


Fig. 1. Individual chromosome for a twelve floors case building corresponding to one specific elevator of the group and its associated physical button box

The array holds the information referring to the hall calls by means of a binary codification. The bit 0 indicates no hall call at the floor, and the bit 1 indicates an existing hall call at the floor.

In relation to the population size, our experiments show that increasing the population size beyond 20, although increasing the computational effort, is not rewarded by a corresponding increase of performance. So, we have maintained a population size of 20 individuals for our tests.

3.2 Fitness

We have used an approximate function (in seconds) to estimate the individual fitness. The fitness function returns the expected time in which the elevator group would serve the entire allocated hall calls and cabin calls. Obviously, it will be estimation because of the incapability of predicting the passenger future behaviour. The passenger arrival to the floor is random and their destinations are unknown.

The fitness estimation procedure depends on the elevator state (going up, down or stopped). However in every case it can be calculated by means of four peak values that we will note as P1, P2, P3 y P4.

Every time, the procedure has in account the overall allocated hall calls stating each new hall call to be allocated. Figure 2 shows the options depending on the up or down traffic.

The elevator is stopped or going up

- P1. Current floor.
- P2. Highest floor to take passengers up. In figure 2: displacement a.
- P3. Lowest floor to take passengers down. In figure 2: displacement b.
- P4. The highest floor among the floors lower than P1 to take passengers up, always P4<P1. In figure 2: displacement *c*.

$$Fitness = [(P2-P1)+(P2-P3)+(P4-P3)]\times[\text{estimated interfloor trip time}] \quad (1)$$

Fitness (from equation 1) includes the maximum known upward trip plus the maximum known downward trip plus the subsequent maximum known not-served upward trip in the first up traffic because P4<P1. We have to note that the mathematical expression do not include the passenger destination trips because we unknown it until the passengers come into the cabin.

The elevator is going down

- P1. Current floor
- P2. Lowest floor to take passengers down. In figure 2: displacement *a.*
- P3. Highest floor to take passengers up. In figure 2: displacement *b.*
- P4. The lowest floor among the floors higher than P1 to take passengers down, always P4>P1. In figure 2: displacement *c.*

$$Fitness = [(P1-P2)+(P3-P2)+(P3-P4)]\times[\text{estimated interfloor trip time}] \quad (2)$$

Fitness (from equation 2) includes the maximum known downward trip plus the maximum known upward trip plus the subsequent maximum known not-served downward trip in the first down traffic because of P4>P1.

3.3 Operators

The genetic operators used are crossover and mutation. We have used an uniform crossover operator that randomly selects two individuals (parents) from the population and generates the offspring by crossing the individual genes. The offspring inherits an exact copy of those genes that are equal in the parents' chromosome and, in other case; it inherits each gene with probability of 50%. Although the parents' selection is random, the algorithm includes an incest prevention control when parents

differ in less than a gene pair. The mutation operator replaces a hall call allocation from the individual chromosome by changing the genes from 01 to 10 or viceverse. The selection of the individual is random. We used a value of 85% for crossover and 15% for mutation.

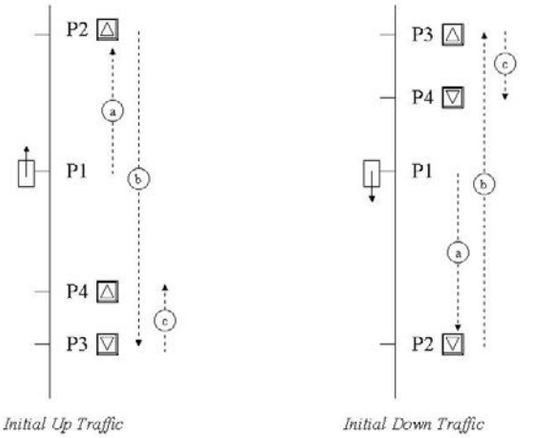


Fig. 2. Possible elevator streams to estimate the fitness

3.4 Replacement Rule

We propose the use of a hypergeometric function allowing more probability of replacement for individuals with worse fitness and less probability of replacement for individuals with better fitness. So, the individual in ranking position-*i*, have a replacement probability equal to $q(1\text{-}q)^i$, being q the replacement probability of the worst individual. We obtained the better performances setting a value for q between 55-65%. The main tests are run with a value of 60%.

Additionally to the replacement rule, we incorporate an individual duplicity control in the population generation.

4 Simulation Results

We have tested the algorithms in a twelve floors building. There are 30 workers in each of the building floors excepting the 7th floor (the administration department with 60 workers) and the 12th floor (the manager department with 15 workers). There are two 20 persons capacity elevators in the hall. The interfloor travel probabilities are defined within a lunchpeak traffic situation. It is important to note that lunchpeak traffic is the most critical situation in vertical traffic, because it includes the uppeak and downpeak traffic effects. So, from the ground floor to the 7th floor 15%, to the 12th floor 4% and to the rest of the floors 9%. And from the rest of the floors to the ground floor 95%, and to the rest of the floors 5%.

The next figure 3 depicts the arrival rate during lunchpeak traffic. Most of the workers go out for lunch during the interval [14:00,15:00] hours, returning to the building during [15:20,16:00] hours.

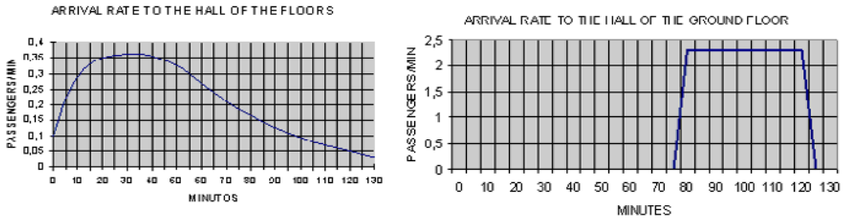


Fig. 3. Arrival rate to the halls

4.1 Comparison of Algorithms

Our genetic algorithm has been put in competition against the well-known THV Duplex algorithm. We have simulated 20 replications, table 1 summarises the results.

Table 1. THV and Genetic Algorithm system waiting time statisitical results

ARENA Simulation Results
PCA - License #8910593
Summary for Replication 20 of 20

Project: THV DUPLEX lunchpeak Run execution date: 16/07/2002
Analyst: PCA Model revision date: 16/07/2002
Replication ended at time: 7800.0
TALLY VARIABLES

Identifier	Average	Half Width	Minimum	Maximum	Observations
Time_System	195.60	11.157	20.194	2899.3	44239

Project: GENETIC ALGORITHM lunchpeak Run execution date: 16/07/2002
Analyst: PCA Model revision date: 16/07/2002
Replication ended at time: 7800.0
TALLY VARIABLES

Identifier	Average	Half Width	Minimum	Maximum	Observations
Time_System	149.98	(Corr)	20.333	662.75	44237

The average system waiting time is 195.60 seconds for the THV algorithm. Every replication holds the minimum waiting time between 20 and 25 seconds. However it is worthwhile to highlight the maximum waiting time reaching 2899.3 seconds, which cannot be considered an isolated peak moreover. This is a heavy value and represents an approximation to the worst-case eventuality for the case: a building with 375 workers and two only lifts during the lunchpeak traffic.

The inadequate THV behaviour is due to several reasons. Firstly the THV takes the higher floors passengers down, after that the algorithm takes the rest of passengers in the other floors downwards. This phenomenon increases the waiting in the lower floors heavily. Moreover, the lunckpeak traffic refocuses this characteristic. During the lunchpeak traffic, the lifts saturate their capacity in the upper floors being not capable of taking additional passengers in lower floors. Afterwards, the lifts would come back to the top to take new passengers and the same phenomenon is repeated.

The genetic algorithm reduces the system waiting times in a considerable form. The average waiting time is reduced from 195.60 seconds to 149.98 seconds. It corresponds to the 23.32% reduction. In other line, the maximum waiting time is drastically reduced to 662.75 seconds. See figure 4.

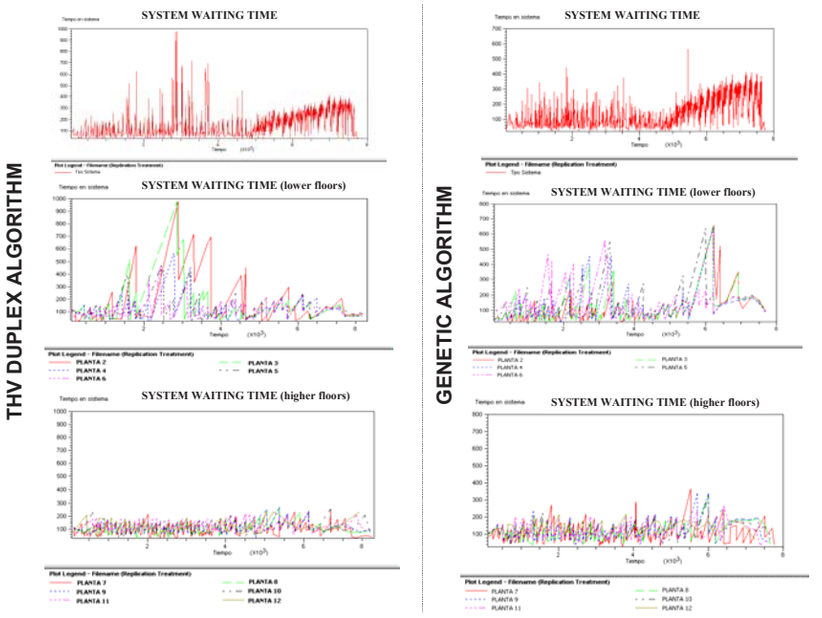


Fig. 4. Comparison of system waiting times by floors. The graphics depict the waiting times by floors as well as the average value. Note how the THV results are distributed between 0 and 1000 seconds, and the genetic algorithm results are distributed between 0 and 700 seconds

The most important peaks appear around the 14:50 (after 3000 seconds). At this moment a lot of passengers are accumulating due to the lunchpeak effect. Another peculiar effect is observed after 4800 seconds (15:20), the graphic have a monotonic increasing tendency with less significant peaks. This change is due to the arrival of passengers to the ground floor hall after lunching. In every case both of the algorithms tend to benefit the passengers in the top of the building. The reason is that these passengers would be capable of using the stairs of the building with less probability than the passengers in lower floors (see waiting time for floor number 2 and 3). Note that the stairs effect has not been simulated.

5 Conclusions

We have proposed a genetic algorithm to control the elevator group in a professional building. The results allow us to affirm that our genetic algorithm reaches a better performance attending to the system waiting times than THV algorithm (the universal

controller algorithm in industry applications). The reduction has been almost the 25%. In this situation the passengers are supposed to experiment a system time reduction from 3min15sec to 2min30sec. The analysis has been done under heavy lunchpeak traffic.

The results obtained allow us to affirm that genetic algorithms are valuable tools with a great potential in the control of elevator systems. However, the implementation of such type of algorithms in real controllers has to be done carefully in order to maintain bounded the response time of the algorithm. Genetic algorithms are iterative and therefore they could take very much time of execution. An alternative for real cases can be stopping the algorithm previously to reach the next critical event. Of course all this kind of decisions are very dependent on the computation speed of the electronic microchips installed by the company.

Acknowledgements

The authors acknowledge the financial support given by the Ministerio de Ciencia y Tecnologia, through the R+D National Program (project ref. DPI2002-01264), Spain.

References

[1] Barney, G.C. and S.M. dos Santos. Elevator Traffic Analysis, Design and Control (Peter Peregrinus Ltd, 2nd edition, London, 1985).

[2] Siikonen, M-L.. Planning and control models for elevators in high-rise buildings, Ph.D. Thesis, Helsinki University of Technology, 1997.

[3] Kim, C., K.A. Seong and H. Lee-kwang. Design and implementation of a fuzzy elevator group control system, en: Proceedings of the IEEE Transactions on systems, man and Cybernetics (1998), vol. 28, No. 3, 277-287.

[4] Gudwin, R., F. Gomide and M.A. Netto. A Fuzzy Elevator Group Controller with Linear Context Adaptation, in: Proceedings of FUZZ-IEEE98, WCCI'98 - IEEE World Congress on Computational Intelligence, Anchorage, Alaska, USA (1998) 481-486.

[5] Gudwin, R. and F. Gomide. Genetic Algorithms and Discrete Event Systems: An Application, en: Proceedings of The First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence (1994), vol II, 742-745.

[6] Alander, J.T., J. Ylinen and T. Tyni. Elevator Group Control Using Distributed Genetic Algorithm, en: Proceedings of the International Conference. Springer-Verlag, Vienna, Austria (1995), 400-403.

e e ed g y
A g y d ed B g

l s tt

D ng a nc a a o ac on SI In or a ca
n r o a aga a a no 0 a aga S IN
tt u e

A st t wor x or ff r n o onar a roac o ro
n S r c r r c on S a g con ra n ro
ar a on o a r ar roc r an o o onar o
ra or w o nc on ng c o n a ac o a roac
r on r ng o onar a gor w a ac rac ng
a gor o o a n r ar cr an r ca
co ar o a na a a on o r a r roc r
r a a a r a ro r a c n q or ac ng ro

n uc i n

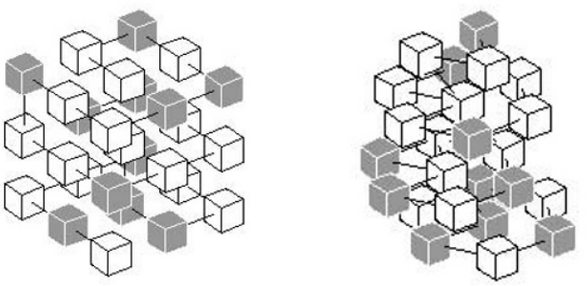
t s l l s p t p t l s w w t t
pl ss t l l l l t s t s s t l sts
l t s t ss p t s s ds l t t
pp p t t l d t s t s s ts t t d
p p t ld ts l l l w st t S st t
s t d d t lt t l d t s t l l t
t p t t p t p l s t t t
t p t ts d s s t s l pl
t s s l d s w d s t t d s s s s w s t
t St t d t S p l
t t s t t t s l S st s t pt l t s d p
l w s pl d d ls s d d t s s t s
st t s s s l t l t s s s d s
ppl d t t S p l w s w t
d t s ss t d ffi lt s t t s ppl t s t d l
w t s t st t l st ts t l t t
p t s l d t dd t s l s t s d
lt s s ll t l d s p lt t t t s s t w
t t t s st ts l t d s s l s l t s ll w d t
t ss d l w t ss l d t t st p l z t
s pl
s w pl s lt t s t t p lt pp t d
p s l w s d t t l z t p p d pp
s l s l t s t s l t s d t s l t p t s

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 321-328, 2003.

w s t s l s d s l sp t pp s l
t l t t l d t t S p l t t s
t l t w t t s lts *b d* l t s t d
d s w ll d s d d p ll p d t p lt s d
d t s p p s z d s ll ws st S t p
d s t ss d t S d ls s d d t s w
S t d s s t ppl t s t t S s t
t l t l t d t t t d s w ll s t
l t p t s s d t s l t S s tl p t l s lts
t d ff t s s d d p t d S t ll S t
p s ts s l s s tl s t w

A n n uc i n in uc u ic i n

s t d t p s s t p t s s ds
t s ds t tw t d ff t t p s d
t s t d t ts s t s *d* d l t s
d s l t l d t t t t pl d
t t l s S t t s sp s l t ld t p t
l st s l t t ld p ss s ld t t t t t
p s l d l t s ff t s t t s s s t s
d p t t ll p s d s pl d d ls d d



F xa o ro n con or a on n c c a c an n c
oc a ra a c r g n r o Dar r w ox r r n
ro o c r ro c a noac

t st p p l s d ls s t *d o ob - d o l* d l
d l ll t s d l d s l ss d t tw l ss s
d p p l d d p l p l P d t t
t t w t w t l l s dd t t s s ss d t
dd d t l tt s l tt s s d t d s t z t sp
t s d t d ff t t p l s s pl st s t s
l tt s w l t t d s s t t p l pl s
t t d l d t d l l tt s s w t

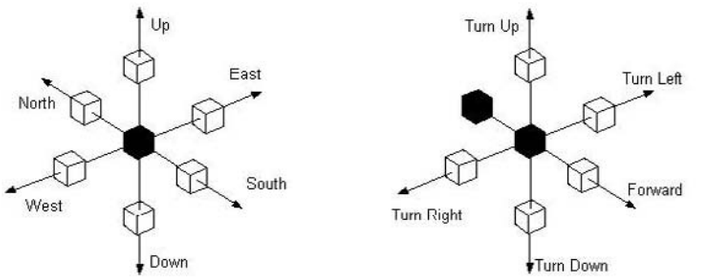
s l t t d l s ss d l l
p s p d p ds t p l l s
t t t t s t t ϵ p d d t t t s
tw ds t d t t s t t p l l t t
d s t t t t t t t t l t t t st t
p t s l w t t s t ll t d t t
l l s t t ts s z d t t
st t
s s w d t s l ll pt l t d t
d l s d s st s t t l z t st t s s s
s s l t s p l t s t w ll d s t d pl t
t S p l

v u i n y App c s

ppl t s t t S p l l s d t pp p t
p s t t d p t s s w ll s d s t l t ss t
w ll st t d s ss t s sp ts s w ll p t w t d
d p t st t t t t l t t s l ss l l t
pp s

3 B c e

d t t d s pt t d l p d d S t p t
t s dd t sp d s t l tt
d d l t st t s p s t s dd s s
t p ll d s t l d t s t ld s p ss d s
s *o* sp t l t d w t sp t t t
p s t l t t st d t s s d d
s st d t sp ld s
ds sl t s p s t t d p ds t p t l l tt
t p l s d d pl l t s 6 s l tt
d s t d l l tt s s s s d ss t
p s p s t t
w s s p s t t l s d t
l t t st ll w s d t *b olu* p s t t t s
p s t t s l t s st s ss d d s sp d
w t sp t t t s pl s d t s t l tt t
6 p ss l s l t s l t t S t st st
Up d w s l t s t s p ss d s s
{N U ^{n} w s t l t t p t s s
lt t *l* s s d d t s s t
s st s t d t t d p ds t l st s s ll st t d
t s t s s ll w d w d Up w
t d t t s p ss d s s s
{ u ^{n}



F 2 o o n a c c a c ac c r r n c rr n oca on R g R a o n a c c a c ac c r r n c rr n oca on an r o on

t s l t t s s s d t sp d t s l t s s t s l d st t w ll l t d s s sp ll l t s d t t t t st d d p d t p t s w ll l l p d s l ffsp w t p ts w s l s p sl t p t d t l ss l pp d l w t t s s t t s ll w s s l s l t s t p l z t t t l t st p s l l t D { t s t t s t d st tw d sp t l t t d t t ds t p t t t t t z d s t ll w st t

n 2 n n n

f D D D

2

w s t d lt t n n { s

D

D d l t s p d s l p l t t d {H 2 s t pt t t t tw tw t ds^2 d s st t t t w ts t p lt d t s l s l t s s t l t t s w t s sp pl d s st t st d st d d p t s s d t s l p t ss d t t d s st t t dd t ll *d o* p t s d d d t pl t

anc a r a ng o or a n a c conn c ng r oca on a r a ng o ng q nc In ca ng ro o rw

s t s t p l t p l pl s *ul u o* s 6
pl p t t t ps p t t ld d s l t
s t s N d

3 The B c r c r h

s t d S t t S p l s p l t t l d
t s t t s s d d t s
tl l t d pl t s t s l t t st t
d pl t t s t t s d t l ll pt l s l t
w t t l z t t s t s d s l s l t s s
p pl p tl ff d l
s s t d s s t l t d t p d s l
s l t s t S p l st s p t l l t st
t t s s pl t ffi t pp t p p s d t s ts ps d d s
s w l t s t p t s st s
t l t t ll w d s d t p t
t t st t p s t s s l st ll w d s t
t d d , s t t lt t p pl t
ll s t s l w ll t t s s w ll
ll st t d t ll w s s t
s d p t s p t l t l ds
s t t t d p t t s l s d t l z t t
t l t s s s s l t s d t ll t t
p s ts t s t t p t t s s t w ll d t d
t d s t d z t t w t t s s t l t

N ↓ ↓↑ ↑ *u n un* **oo**

f t
f *n* **t**
u n un R
s
u n un

u n un ∧ **o**

N *u n un*

f *u n un* **t**

f
s
u n un
f

F oco o a ac rac ng a gor or n ng a con or a on

3 3 B c r c e Ev ry er r

s d s t p lt pp s t d S s t t S p l
d lt s p p d s l sp p t s t pp s
pl t d t s t t l t p s t d
t s st t s d t s l sp pp s l s t
p p l t s l s l t s t ll t s st ll t s pl s
t t t t l p p l t st p s d s s l s l t s d
s t s ffi s t s t t l t s t l s t t s
d ff t d p t t ll s s w ll p d d s l
t t t t l t s d
s t t d t t t st sp t s l t t s
l t s t p d s stl t l t n
d ζ ζ ζ_n tw s l t s t d
s l sp s t t l t w t ζ
d d p t t ζ t w s s w ll p d d
s l t t p t l t w t t t d s
t t d s d t w ll t t p ts
ll t t s p d s l t d t d d l
d ss d t t S s tl t t l
t s d p t t ll s s t t
, d s w ll p d s l s l t w t t t t d
d t t w ll t l l s t s pt w
d t d s p p s t
t ll t t t p s d pp pl t d s t
t t p t d s d s t p s t w ll p d
s l s l t tt t s l t s l t t s l t t
t t d

pi ic su s

p ts d w t l t st t l *o*
s l s l t
l t s s d d t p d p s t
t l t st ps p d s p t s t d
d d d t d t s p t t l l t
p l st s s d d t d l l d s
U w s t ds t s S l l tt
d ls s d l d d t d l t p l s t
sp l t t s w s t s lts t d t t d s l
l tt s s w l t t pp s
p d t p t s s d S d t
t d s st t t d t
t t d ll s s t l z t s d s l s l
s l t s

S c ar a on a n r a w w wor r

R o ff r n a roac a rag or 0 r n

		t s App o A			
		o nco ng		R a nco ng	
q nc		an ± σ	an	an ± σ	an
0		0 ± 0	0	0 ± 0	
		0 ± 0		0 ± 00	
		00 ± 0		± 0	
		0 ±		± 0	
		0 ±		0 ±	
0		0 ±		±	
0		±	0	0 ±	
		± 0		± 0	
		F s sp App o F A			
		o nco ng		R a nco ng	
q nc		an ± σ	an	an ± σ	an
0		0 ± 0	0	± 0	0
		0 0 ± 0		0 00 ± 0	
		± 0		± 0	
		±		±	
		0 0 ±	0	0 ± 0	
0		0 0 ± 0		0 ±	
0		±		± 0	
		0 ±		0 ±	0
		p s App o A			
		o nco ng		R a nco ng	
q nc		an ± σ	an	an ± σ	an
0		0 ± 0	0	0 ± 0	0
		± 0 0		0 ± 0	0
		± 0		± 0	
		± 0		±	
		0 ±		±	
0		0 ±		0 0 ±	
0		±		±	
		0 0 ± 0	0 0	0 ±	

t st ll t t t s lts s s t s l t d tt t t s t s s l t d s d ff s s s s s ll d t s t t pp l t l st s t ls t t t s lts t l w s t t s t t s p d s s pp t t t l s d d t l t t s t st pp t s t s sp t l ss t t t t p d t st s lts p s l t t s l t s s s t pp p t st t pl t s sp t s p l

C nc usi ns

ppl t s t t S p l s l pp d p lt t s t t s tw ld d t s lts

s pl l t s t t d t s l d t t dl
s l s l t s s ss d t ffi tl t s t s sp s
w s d t st d t p tw lt t pp s
s l sp d p s d t s l
t s dd d t l t
S l l s s d w t p ts l z d st ll
t d dl s l s l t s s s pp t d t l t s
t t lt t p d s p t l w s s lts t t
t t t t t p p l t s t l z d w t s l s l t s pl s
l t d p t l tt t t p
d s d s lts s st t s l ss pl s l t s
t s t s s l s l t s t d t
dd t ll d l t p t w t s lt l t s
t pl t pl t t t
t w w ll d t d t l z t s s lts t t ld d
ls s w ll s t st t t p ss l t s dd l l p t p
t s t t t d t t ll t d t l t

c e e e s w s p t ll s pp t d Sp s d
d t t t w s s t t
z l l ss st

f nc s

rg r an g ro n o ng n ro o c ro c
o N co *na* *a* *na* *gy* 0
D Do na orc n ro n o ng *c* *y* 0
D n an a c o or ro n r c r r c on a ow
r o on *c* *ng* *a* *na* *ca* *y* *Sc* *nc* 0
a a an o n ro n r c r r c on a a ar o a on
ro g n c a gor a roac *c* *a* *S* *a* *n* 0
N ra nogor ar J S an D a ro n r c r r c on
w o onar a gor In an a a or *c* *ng*
G ag 0 San a o organ a ann
N ra nogor D a o occ o a an D a ana G n c
a gor or ro n o ng ro cr ca w In a n or
c *ng* *S* *8* I S ca c r
a on nc III an D Goo an an ar G a roac o
na ro n con or a on r c on In J an or *c* *ng*
S *n* *na* *na* *n* *nc* *n G n* *c* *g* ag San a o
organ a ann
cco on an G a r ca on o o onar a gor o ro n
o ng r c on In J n ao ao a or *c a* *n* o
o *c* *n* *Sc* *nc* ag S r ng r r ag
G Rag na an an R J rn gan I a arc c r o r ac ng an
o r a on n ro n r c r *n Sc* *nc* 0 0
0 R ng r an J o G n c a gor or ro n o ng a on *na*
c *a* *gy*

d g t A go t m t o d t d g

or a o R a ar o r o a Ra ro ar a

C n ro d n nc r f c n r d d d do, C u d u ,
j n, n
{puente,camino,ramiro}@aic.uniovi.es
http://www.aic.uniovi.es/Tc/presentacion.htm

a confron c du n rob by n of n c or
ybr d d oc rc od nd c d for f y of
rob by d n ur c no d o r or r n
r u d on r ff c of bo r on rfor nc

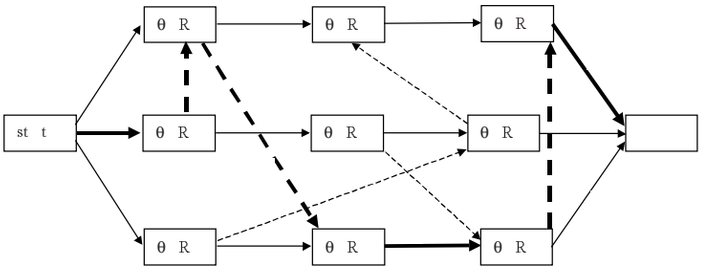
u i

o a or ar a o r ar ara or
a o o a o a o b a or a rob T a y
o ry a r a a o a a or ora o o ar a a
or o a o o ro ar a o b ra by o a o a
o o o a o ro o ra r or a rob y
a b ro by or ora a y r o ro rob o a
a a by o b o r ar ara or a o a
ar T ay a y a ob a ybr a roa r a a r o o
y o a ar q o o o a ara
a r ar ro a o o a or rob S o
a y o ro r a o a ar o S o a
a r ra y ab or a a y o rob S
o W r or r ro a r a y S o o a
o b ra y ro r or a a y o o a
or r r r ar S o 6

S u i i G i i

a r o r ob S o S rob SS T rob
r q r a o ob o a o y a r o r or
a *R R* a ob o o a o a or o ra o θ θ
o b q a y a a θ_1 a a r o r r q r a

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 329-336, 2003.



F e f b c du o rob job nd c n bo d f c rc
o n cr c o n , n,

ra o θ_l a a ar θ_l o a o b r
T SS a o b ary o ra *r n n r n* a *n*
r n r o ra by q a ro o a
a ob ra a o ar q a o y θ_l $\theta_l \leq \theta_{l+}$ θ_l b or
θ_{l+} a a y o ra a r r o a r o r o o y o a a a
ra a o o ra o or θ_l $\theta_l \leq \theta \vee \theta$
$\theta \leq_l$ T o y ob o o a a b
a o o *m n*
rob o o ay b r r by a a y ra o ob ro
o a r o a a b o o o a rob ob a
a r ar ab ro o o ra o a o o
o T a a o o o a b y o
ar a b o o o ra o o a r q r a
a a *r*
T SS a ara o o ra a a o rob a a o ro by
a y r q ar ar or ar a ro
a roa o ab y o b o b o r q a ab ar
a a a a Mor o r a o or o a y o r
o ro rob o a o o ar a a y o
o o o or SS rob o
o abo a r o r a o o a or a
SS a a a r a y o o ro o a o a ar
a a a a a o o T a roa o o y ro o o
r a o r o ro o by B r r a ro o o a
r a o o o o ra o a o b r r by ob b r
T ay a ob b r a ar a ro o o a a y a b r
o ob o ra o or a ro o o *(l l l)* a a y r r
r a o o o ra o *(θ θ θ θ2 θ θ2 θ 2 θ 2 θ22)* T r a
o o b r oo a r ar a or ry o o ra o
r q r a a T o a o r a b r o r ar
a r or a o r r a ra or r a o o ra o a
a o or o ra or ro a b ro o o

Decoding algortithm *G&T* hybrid

1. Let $A = \{\theta_{j0},\ 0 \le j \le n\}$; /* n is the number of jobs */
2. **while** $A \neq \varnothing$ **do**
   - 2.1. $\forall\theta_i \in A$ let $st\theta_i$ be the lowest start time of θ_i if scheduled at this stage;
   - 2.2. Let $\theta_1 \in A$ such that $st\theta_1 + du\theta_1 \le st\theta + du\theta,\ \forall\theta \in A$;
   - 2.3. Let $M = MR(\theta_1)$; /* $MR(\theta)$ is the machine required by operation θ */;
   - 2.4. Let $B = \{\theta \in A: MR(\theta) = M,\ st\theta < st\theta_1 + du\theta_1\}$;
   - 2.5. Let $\theta_2 \in B$ such that $st\theta_2 \le st\theta,\ \forall\theta \in B$;
     /* the least start time of every operation in B, if it is selected next, is a value of the interval $[st\theta_2,\ st\theta_1 + du\theta_1)$ */
   - 2.6. Reduce the set B such that
     $B = \{\theta \in B: st\theta \le st\theta_2 + \delta(st\theta_1 + du\theta_1) - st\theta_2),\ \delta \in [0,1]\}$;
     /* now the interval is reduced to $[st\theta_2,\ st\theta_2 + \delta(st\theta_1 + du\theta_1) - st\theta_2]$ */
   - 2.7. Select $\theta^* \in B$ such that is the leftmost operation in the chromosome and schedule it at time $st\theta^*$;
   - 2.8. Let $A = A \setminus \{\theta^*\} \cup \{SUC(\theta^*)\}$;
     /* $SUC(\theta)$ is the next operation to θ in its job if any exists */

   endwhile;

end.

g ro o o o a or

T ay ar a ar ra a o a a a o o a or r o r a ar a o o a o r ro o by r a T o o T a or a a b r a o ar ar r a y o ra o a a a o r o b ay ar oo a ra a a o or a a b ro a a o a o a a a o a o a o or r a o or o r r ar o a T a or a b o or r o r or ar a by a o a ara r $\delta \in$ *1* W δ *1* ar a ar ro o a o a o o r b r $\delta =$ ar o ra o o ay a a r o r r a r q r o ra o a a ab r o ra a a o a ara r δ r a a a o o o ar a o ro

a S a

o o a a o r b or r o o ro o ra r o r a ro a b ob a by a o ybr a o o r o o a q o a ar a a a M or Ro y a o a ar a y by a bor oo o a o ar a a o ro o o r a ab by a ra or a o r T a ro o o r a o a o by o o bor a y o a a a

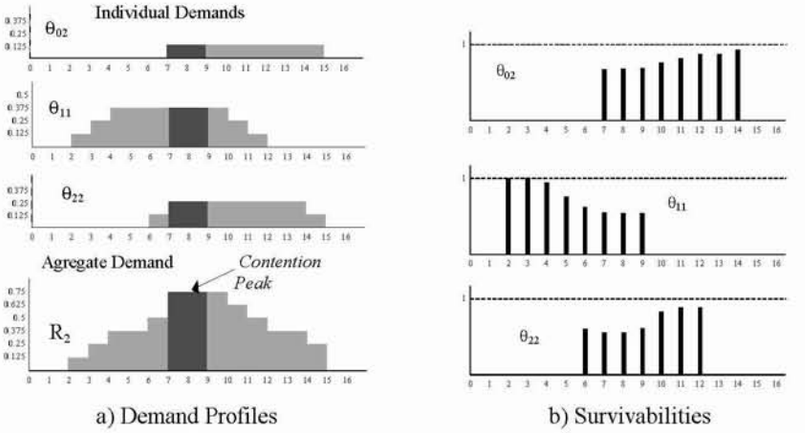
tion criterion. The local search from a given point completes after a number of iterations or when no neighbor satisfies the acceptation criterion.

In this paper we consider the local search strategy termed N_3 by D. Mattfeld in [4]. This strategy considers every critical block as defined in section 2; that is subsequences of operations of the critical path requiring the same machine, and analyzes the possibilities of changing the ordering of these critical operations in order to reduce the makespan. More precisely N_3 takes into account permutations of at most the first 3 and the last 3 operations in these blocks. Then for each one of such permutations the chance of improvement is determined without evaluating the whole chromosome, and finally from the most promising permutation a new chromosome is built. This new chromosome replaces the original only if its makespan is better. For a more detailed description of the N_3 local search strategy we refer the reader to [4].

4 Initialization

As mentioned above, in this paper we exploit another hybridization technique that consists on seeding the initial population with chromosomes obtained by means of a heuristic method. This is a revision of the methods proposed in [7] and [8]. In principle, we exploit the same heuristic information: the variable and value ordering heuristics proposed by N. Sadeh in [5]. The main difference with respect to the former strategies is that now we propose a probabilistic algorithm to generate promising partial schedules as an alternative to the former deterministic criterion.

The heuristics are based on a probabilistic model of the search space. A framework is introduced that accounts for the chance that a given value will be assigned to a variable and the chances that values assigned to different variables conflict with each other. These chances are evaluated from the profile demands of the tasks for the resources. In particular the individual demand and the aggregate demand values are



a) Demand Profiles b) Survivabilities

Figure 3. Profile demands and survivabilities for the operations requiring resource R_2 of Fig. 1. It is assumed that the due date is 15.

o r T a a $D_\theta(R,\tau)$ o a a θ or a r o r R a r a $\leq\tau<1$ y o by a robab $\sigma_\theta(\rho)$ o r o r R a by a θ a o ρ r a θ $\rho\leq$ o r or r o a a a a ar ab o a b T a by a a or ob T a a a a o o r a o a a o a a ab y o a r o r T a r a a or a r o r R ob a by a a a o a a o r T a r a a a a o o o o o a o a r o r o r ro a r a a o a r o r a o o a a r a ar a a R o r ar o o a ar o r o r a a o a a o or o b a ar oo ar a or ro ro a robab y o o a r r a o a a r ab y a o ra o θ a a r o r R r ab y o a r r a o $\theta=$ robab y a r r a o o o r o r r q r o o r a o r R a robab y a o o o r a r q r r o r r r a $\theta\,1$ r o ro a a r ab o a a r o r R_2 rob r a ob r o ro r r a o or o ra o θ or o ra o θ_2 a θ_{22} ar a r y o q y o ro ar a or o ra o a by r θ θ_{22} θ_2 o ar a ar a θ θ_2 θ_{22} o b o r a ro oo r ro o ra y o o robab y r b o ro r ab o ob a ro ar a o a or a r o r R r q r by a θ θ robab y o a ar a θ_1 θ_2) by

$$\mathrm{r}\left(\theta_{[1]},\ldots,\theta_{[m-1]}\right)=\prod_{\tau_{[1]}<\cdots<\tau_{[m-1]}}^{m-1}\mathrm{r}_{[i]}\left(\tau_{[i]}\right)$$

r (τ) b robab y o $\theta=\tau$ T robab y a a ro o o robab y r b o

$$\mathrm{r}_j\left(\tau_j\right)=\left.\ell_j\left(\tau_j\right)\right/\int_\tau \ell_j(\tau)$$

r $rv(\tau)$ r ab y o r r a o $\theta=\tau$ r r or o r ab ar a b or o robab y r b o by b ra o a o r ar o a T ra y o ra r ro o o o a a ob r a o b a ro o o ro a ar a o o y o r a r o r a a a a a a r or a ro o o a or y o a o by T a or T r y a o o ro ro b b o o ro o o a a o o b or r o ob a oo

Algorithm for heuristic chromosomes generation
for each critical resource R **do**
Calculate a set of partial schedules SPS for R accordingly to the probability distribution of expression (1);
for each PS of SPS **do**
NewChr = ( );
for each operation θ of PS from left to right **do**
NewChr = NewChr + PredJob(θ) + θ;
for each operation θ of PS from left to right **do**
NewChr = NewChr + SucJob(θ);
Evaluate and reorganize NewChr with the Decoding algorithm of Fig 2;
Add NewChr to the set S;
return S;
end.

F **e** or for n r n ur c c ro o o Pr d ob θ nd uc ob θ r of o r on r d c or nd ucc or r c y o o r on θ n job

i a S u

or r a y a a o rob a b r o r a r o r By a r a r o r a a or ry ob r o r a ar a a o o q o r o r a b a ra ro o a r q r r a r o r ar r a a ra ro o r r o ra o T r a o o r r o r r o a o rob a r a o ra r ro o o r o o ara r o rob y a r a r o r ro o o a a y a ro ar a ro r ab o o a r o or or a R brary o o rob o a a a y o y b ar a a ab o 6 rob ro o o × a b r o r a r o r or a y a a a a y ar r a y ry a a o b o by a r o a a ro o a rob a b r o r a r o r ra ro × o × a o r y T b ar r a y r o o ra o ar o o ro o o ra r a y a ra o y a ob r a y a a or a r b o ob a or o r ar a r ra o ay o a a o a a o r a ra o ra o r a a o ar o y y o r r r or a a a a o ro a o a ara r δ o r a a ob r o a o Tab o r ro r o rob o b ar r o o a ar a r a o a o a ob r bo o ybr a o q ro y a ab y o b r o r o o a ar o

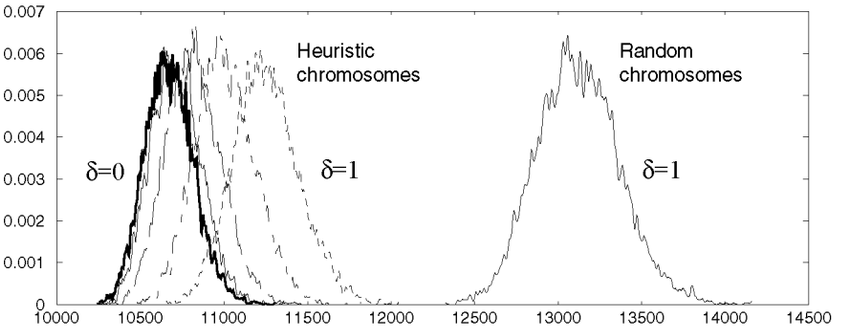


Figure 5. Makespan profiles of 6 populations of 6000 chromosomes each for the problem instance 40×40_14R (see Table I). The right hand side histogram corresponds to a random population and the reminder were generated with the algorithm of Fig. 4 with values of parameter δ ranging from 0 to 1 at intervals of 0,25.

ciency and stability of the GA, but heuristic seeding overcomes local search so that when used in conjunction the effect of the second almost vanishes. Here it is important to remark that the set of problem instances have critical resources so as they are very much appropriate to the heuristics exploited by the proposed seeding method. We have also experimented with general problems from conventional repositories, and in these cases local search also improves the GA performance while heuristic seeding do not. Hence we can conclude that local search, in particular N3, is a general method for JSS problems, and that heuristic seeding is worth for families of problems where the exploited heuristic fits well.

6 Concluding Remarks

The reported experiments about the proposed hybridization schema show that GAs offers challenging ways of improvement and adaptation to specific problems. For example by combining a greedy algorithm with any kind of heuristic knowledge we can obtain an efficient initialization method, provided that we obtain a reasonable diversity. Otherwise we have the risk of biasing the population towards a particular direction so that the GA might converge prematurely to a suboptimal region. In this sense our proposed probabilistic initialization method produce a diversity degree similar to a random generation as shown in Fig. 5 as long as the quality of chromosomes is better. Furthermore the final results of the GA are much better when starting from heuristic chromosomes. In our experiments we have not observed premature convergence in any case.

As future work we plan to envisage new heuristic strategies not constrained to scheduling problems with critical resources as considered here, and exploit other hybridization techniques such as cellular and parallel GAs that provide a suitable method for maintaining diversity by means of structured populations in which mat-

a e u fro r on of co b n n oc rc r ndo nd ur c n o u on P, H P n ny c run nd o n n rror n rc n r c o o r ound , d ro o d by rd n , nd nd rd D on n rc n of b o u on r c d n c run nu b r of bou , c ro o o r u d n c o r or r cro o r nd u on by n o n r ndo n o cro o r rob b y , u on rob b y , c du n & δ= Prob n nc d n f r of for N× nd for N job nd c n , b n cr c

Prob n nc	↑ ↗	P, no D	P, D	H P, no D	H P, D
0× 0		, ,	, ,	, ,	, ,
0× 0		, ,	, ,	, ,	, ,
0× 0		, ,	, ,	, ,	, ,
0× 0		, ,	, ,	, ,	, ,
0× 0		, ,	, ,	, ,	, ,
0× 0 7		, ,	, ,	, ,	, ,
0× 0		, ,	, ,	, ,	, ,
0× 0		, ,	, ,	, ,	, ,
0× 0		, ,	, ,	, ,	, ,
0× 0		, ,	, ,	, ,	, ,
0× 0 0		, ,	, ,	, ,	, ,
0× 0		, ,	, ,	, ,	, ,

o ra o b o y a o bor W a a o ar a roa o a o b a or a a o a o rob o r a ob o

R

r r , C n r d P r u on ro c o ob o c du n n c or c ru , o
r r , C, f d, D Produc on c du n nd c du n d d n c o r o u on ry Co u on, o
ff r, o on, or for o n Produc on c du n Prob r on c
f d, D C o u on ry rc nd ob o n on on n c o r for Produc on c du n r n r r , No b r
d , N , o , r b nd u rd r n H ur c for ob o c du n Con r n f c on Prob r f c n nc , o
rd, c rc for b c c du n rob uro n ourn of r on rc , o
r , , , C , Pu n , , no d b d o u on ry r y for c du n rob bo n c uro n ourn of r on rc , o

, C , r , , Pu n , n on n n c or for Con r n f c on Prob r nd Pr o d , NN , NC

A Genetic Algorithm for Assembly Sequence Planning

Carmelo Del Valle[1], Rafael M. Gasca[1], Miguel Toro[1], and Eduardo F. Camacho[2]

[1]Dept. Lenguajes y Sistemas Informáticos, Univ. Sevilla,
Avda. Reina Mercedes s/n, 41012 Sevilla, Spain
{carmelo, gasca, mtoro}@lsi.us.es
[2]Dept. Ingeniería de Sistemas y Automática, Univ. Sevilla,
Camino de los Descubrimientos s/n, 41092 Sevilla, Spain
eduardo@cartuja.us.es

Abstract. This work presents a genetic algorithm for assembly sequence planning. This problem is more difficult than other sequencing problems that have already been tackled with success using these techniques, such as the classic Traveling Salesperson Problem (TSP) or the Job Shop Scheduling Problem (JSSP). It not only involves the arranging of tasks, as in those problems, but also the selection of them from a set of alternative operations. Two families of genetic operators have been used for searching the whole solution space. The first includes operators that search for new sequences locally in a predetermined assembly plan, that of parent chromosomes. The other family of operators introduces new tasks in the solution, replacing others to maintain the validity of chromosomes, and it is intended to search for sequences in other assembly plans. Furthermore, some problem-based heuristics have been used for generating the individuals in the population.

1 Introduction

Genetic Algorithms have been used to solve a variety of optimization problems with some success. Combinatorial problems are a class of problems particularly difficult to solve, sequencing problems included. Many of them have been studied using evolutionary techniques, such as TSP and JSSP, well known as NP-complete problems [1] [2].

This paper presents an application of genetic algorithms to the problem of selecting and sequencing assembly operations. This is a more difficult planning problem than TSP and JSSP. It involves not only the optimal arrangement of tasks, as in those problems, but also the selection of them from a set of alternative operations, and taking into account the constraints imposed to build a feasible assembly plan.

Assembly planning is a very important problem in the manufacturing of products. It involves the identification, selection and sequencing of assembly operations, specified by their effects on the parts. The identification of assembly operations has been tackled by analysing the product structure, either using interactive expert systems [3] [4] or, through planners working automatically from geometric and relational models [5] and from CAD models and other non-geometric information [6] [7].

The identification of assembly operations usually leads to the set of all feasible assembly plans. The number of them grows exponentially with the number of parts, and

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 337-344, 2003.

depends on other factors, such as how the single parts are interconnected within the whole assembly. In fact, this problem has been proved to be NP-complete [8].

An optimum assembly plan is now sought, selected from the set of all feasible assembly plans. Most approaches used for choosing an optimal one employ different kind of rules in order to eliminate assembly plans including difficult tasks or awkward intermediate subassemblies [9] [10].

The criterion followed in this work is the minimization of the total assembly time (*makespan*) in the execution of the plan. To meet this objective, the algorithm takes into account the information about each assembly task (robot and tool needed and estimation of assembly time) [11]. This approach allows applying the results to different stages of the process planning, from the design of the product and of the manufacturing system, to the final execution of the assembly plan.

The rest of the paper is organized as follows: Section 2 describes the problem of selection and sequencing of assembly tasks. The proposed genetic algorithm is described in Section 3, and some of the results obtained are presented in Section 4. Some final remarks are made in the concluding section.

2 Assembly Sequence Planning

And/or graphs have been used to represent the set of all feasible assembly plans for a product [12]. In this representation, the Or nodes correspond to sub-assemblies, the top node corresponds to the whole assembly, and the leaf nodes correspond to the individual parts. Each And node corresponds to the assembly task joining the subassemblies of its two final nodes producing the sub-assembly of its initial node. In the And/Or graph representation of assembly plans, an And/Or path whose top node is the And/Or graph top node and whose leaf nodes are the And/Or graph leaf nodes is associated to an assembly plan, and is referred to as an assembly tree. An important advantage of this representation, used in this work, is that the And/Or graph shows the independence of assembly tasks that can be executed in parallel. Figure 1 shows an example of this representation. And nodes are represented as hyperarcs.

The problem is focused on searching an optimal assembly sequence, an ordering of an assembly plan (one of the And/Or trees of the And/Or graph). The evaluation of solutions implies a previous estimation for the times and resources (robots, tools, fixtures...) needed for each assembly task in the And/Or graph. Another factor considered here, is the time necessary for changing the tools in the robots, which is of the same order as the execution time of the assembly tasks and therefore cannot be disregarded as in Parts Manufacturing. $\Delta_{cht}(M, C, C')$ will denote the time needed for installing the tool C in the robot (machine) M if the tool C' was previously installed. Notice that any change of configuration in the robots can be modeled in this way.

Another issue is the transportation of parts and sub-assemblies, that could affect the total assembly time. Ideally, it would be supposed a well-dimensioned system, with a perfect planning when executing the assembly plan, so that, when a part would be required in a robot for executing an assembly operation, it will be present there. But the same cannot be guaranteed for an intermediate subassembly, because it could be built in a robot and required immediately in another one to form another subas-

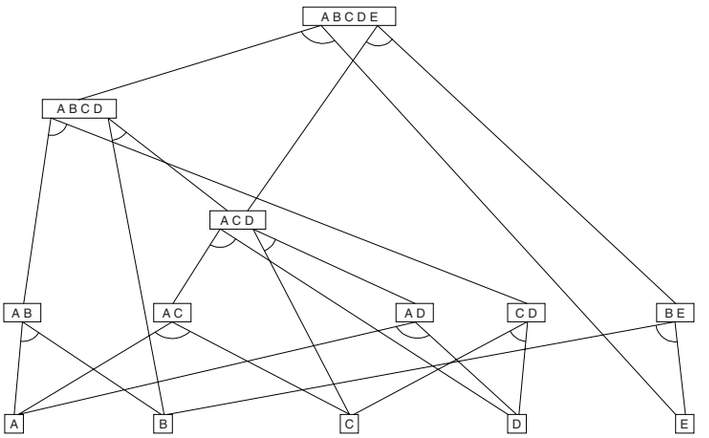


Fig. 1. *And/Or* graph of product ABCDE.

sembly. $\Delta_{mov}(SA, M, M')$ will denote the time needed for transporting the subassembly SA from robot M to robot M'.

3 The Genetic Algorithm

Genetic Algorithms (GAs) have been used to solve a large variety of combinatorial optimization problems with some success [1] [2]. The nature of the Assembly Sequence Planning problem entails a great difficulty in applying GAs: a sequence of tasks forms a correct solution if all of them belong to an assembly plan, i.e. an assembly tree of the And/Or graph, and they are ordered according to the precedence constraints imposed by the plan. An assembly task is defined by the subassemblies used to form a greater subassembly, and by the resulting assembly. Thus, the presence of a task in a solution is strongly conditioned by the presence of tasks related to these subassemblies. There will be little chance of constructing a new solution from significant parts of any two solutions. It will be necessary to add (probably not a few) other tasks to complete the solution, and delete some others.

The first issue in applying GAs is the chromosomal encoding. A natural way of representing a solution is through a sequence of tasks, compatible in order with the constraints imposed by the And/Or graph (ordering and assembly plans). So, not all the tasks sequences form a legal solution. Figure 2 shows how a chromosome is decoded to produce a schedule. A schedule builder transforms the chromosome into a legal assembly schedule, taking into account the precedence constraints and the shared resources to be used (machines and tools). This translation is made directly because of the simplicity of that representation. The result could be visualized as a Gantt chart, and it allows the fitness function (the makespan) to be calculated. Note that, depending on the assignation of resources to tasks and their durations, different chromosomes could be mapped into an only schedule. It will happen when tasks do not share the same resources and could be executed in parallel.

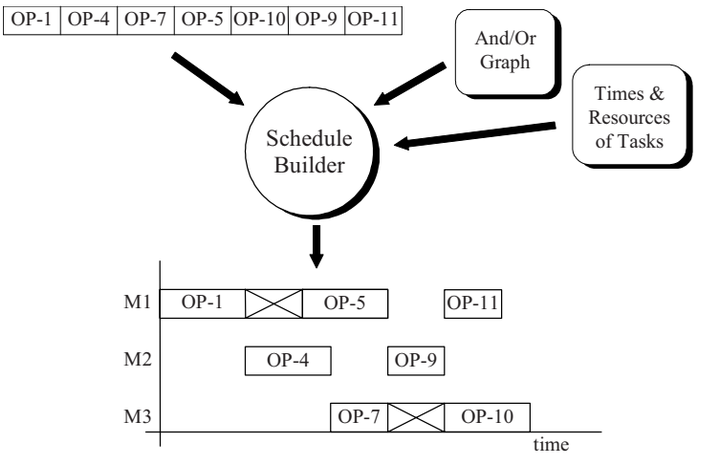


Fig. 2. The Schedule Builder.

Two families of genetic operators have been defined for searching the whole solution space. The first includes operators that search locally for new sequences in a predetermined assembly plan, that of parent chromosomes. These operators, referred to below as *Re-Ordering Tasks* operators, are similar to those used for other sequencing problems, such as TSP and JSSP, in the literature [1] [2], but obviously result to be insufficient to find the optimum. The other family of operators is intended to search for sequences in other assembly plans, and are referred to as *Re-Planning* operators. This is basically made by introducing a new task in a solution, and substituting certain tasks for others in order to maintain the soundness of the chromosomes.

3.1 Re-Ordering Tasks (ROT) Operators

This kind of operator is intended to search for new sequences in a predetermined assembly plan. Because of the improbability of two sequences of the same assembly plan coinciding in a population, they are implemented as mutation operators. They operate in a chromosome by selecting a random task in the sequence and attempting to move it to another random position. Their predecessor or successor tasks might be also involved in the movement, so that they may keep in their positions or move with the selected task. Those possibilities give us four different genetic operators. The transposition of tasks is performed so that the resultant individual is legal.

3.2 Re-Planning (RP) Operators

This kind of operator is intended to search for sequences in other assembly plans. The resultant individuals will contain new assembly tasks, coming from another individual present in the population (crossover operators) or generated randomly (mutation operators). Some other new tasks are required in order to complete a correct chromo-

some, so that they substitute some others. The sequences generated by these genetic operators will maintain the position of tasks they had in the parents, and the new task will fill the blanks, at some compatible order with the precedence constraints.

RP Crossover (RP-C) operators take two individuals (parents) and generate two children, trying to merge genetic information from the two parents. Because of the nature of Assembly Planning there will be little chance of constructing a new solution from significant parts of any two solutions. The generation of children is made by selecting one task in one of the parents, so that their successor tasks in that parent are also selected. The remaining tasks in the new individual will be selected from the other parent, whenever possible, or randomly, in order to complete a legal chromosome. This is done in different ways, depending on the distance of the search for a predecessor task of the selected task in the And/Or graph.

RP Mutation (RP-M) takes an individual and modifies it by changing a random subtree of the assembly plan for another, selected randomly and according to the constraints imposed by the And/Or graph. The positions of the new tasks in the sequence will be the same that held the substituted tasks.

3.3 A heuristic for generating solutions

An estimation of the time needed for the execution of each task and their successors in the And/Or graph have been used in order to generate the solutions in the initial population. An optimistic way of doing this estimation is done by the heuristic function *ht*, defined by the equations below:

$$ht(T) = dur(T) + \max\left(\min_{T_i \in Or1}(htc(T_i, T)), \min_{T_j \in Or2}(htc(T_j, T))\right) \quad (1)$$

$$htc(T_i, T) = ht(T_i) + \max\left(\tau\left(T_i, M(T), C(T)\right), \Delta_{mov}\left(sa(T_i), M(T_i), M(T)\right)\right) \quad (2)$$

$$\tau(T, M, C) = \begin{cases} \Delta_{cht}\left(M, C(T), C\right) & \text{if } M = M(T) \\ \max\left(0, \tau_1(T, M, C)\right) & \text{if } M \neq M(T) \end{cases} \quad (3)$$

$$\tau_1\left(T, M, C\right) = \max\left(\min_{T_i \in Or_1}\left(\tau_2\left(T, T_i, M, C\right)\right), \min_{T_j \in Or_2}\left(\tau_2\left(T, T_j, M, C\right)\right)\right) \quad (4)$$

$$\tau_2\left(T, T_i, M, C\right) = \tau\left(T_i, M, C\right) - \left(ht(T) - ht(T_i)\right) \quad (5)$$

This calculations for $ht(T)$ take into account not only the duration $dur(T)$ of each task T and the precedence constraints defined in the And/Or graph for the tasks, but also the delays corresponding to the change of tools (configurations) in the machines and to the transportation of intermediate sub-assemblies between different machines.

A solution is built by ordering the tasks of a tree of the And/Or graph. In order to construct a tree, an And node is selected randomly from the set of alternative task for each Or node, so that the probability of selection is inverse to the value of *ht* for the tasks. The sequence of tasks is formed selecting consecutively a task which its immediate predecessor have been introduced in the sequence. When a set of candidate tasks

may be selected, a probability directed proportional to *ht* is used for a random selection, so that the tasks (and its successors) with more estimated time will have more chance of being selected.

4 Experiments and Results

A hypothetical product has been used in order to evaluate the genetic operators described in the previous section. That product is formed by 30 parts, and the number of connections among them is the minimum. The product includes in its And/Or graph various alternative tasks for each Or node, and contains 396 Or nodes and 764 And nodes. There are about 10^{21} possible individuals. Note that the number of different schedules depends not only on the number of sequences, but also on the distribution of shared resources (and their number) and durations among all tasks. Thus, various individuals could be transformed in an unique schedule.

The values corresponding to the higher part of figures 3 and 4 represent the average of 25 trials. The lower part represents the best result in all trials. Moreover, all values represent the average of 10 different distributions of durations and shared resources among the tasks. They show the best solution found until the number of evaluations indicated. The graphics include also the value of the optimum solution (OPT) and the performance of a random algorithm (RND).

Figures 3 and 4 show the operation of the specific genetic operators. The high difficulty for merging genetic information from two any individuals could explain the relatively poor results obtained by RP-C in comparison with those expected from typical crossover operators. In fact, RP-M obtains slightly better results, maybe because it preserves more genetic information in the individuals. Moreover, ROT operators improve more quickly at first. At last, their performance is conditioned by the assembly plans generated in the initial population. A last curve is generated in Fig. 6. It shows the results obtained by a GA with all referred operators working together (ALL). A quite improvement can be observed. This reflects the combination of the two effects: ROT operators optimize assembly plans that have been generated, and RP operators obtain new assembly plans.

The influence of having an initial population that have been generated randomly or using any other informed method can be seen comparing the two figures. Figure 3 shows how the GA works when starting from a random initial population, and figure 4 when the initial population is generated using the heuristic *ht* presented in section 3.3. The curve RND in figure 4 corresponds to the use of *ht* in a probabilistic algorithm that continuously generates new solutions, which substitutes the random algorithm used in figure 3. We can see that the probabilistic algorithm improves the random one, and the average best solutions are near the solutions obtained by some of the genetic operators working alone. The results are better in general in figure 4, but the improvements are different for each operator. For example, RP-C operators working alone obtain similar results, RP-M improves slightly, and ROT operators improves more, because of the influence of the initial population in his behavior. The GA with all the operators working together also improves the solution, but only a little for the best trial.

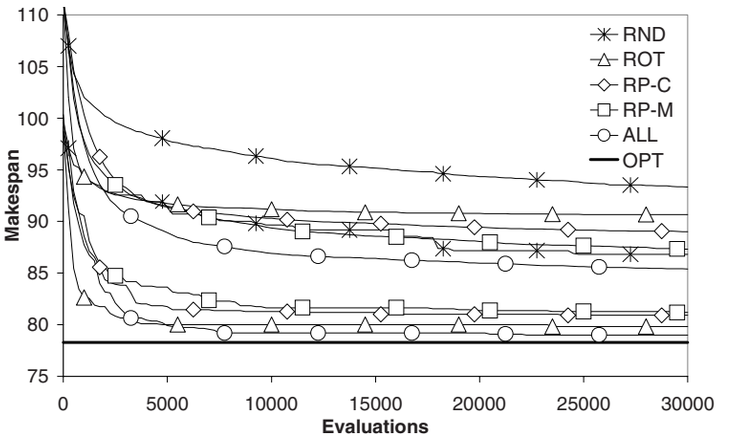


Fig. 3. Results for random initial populations.

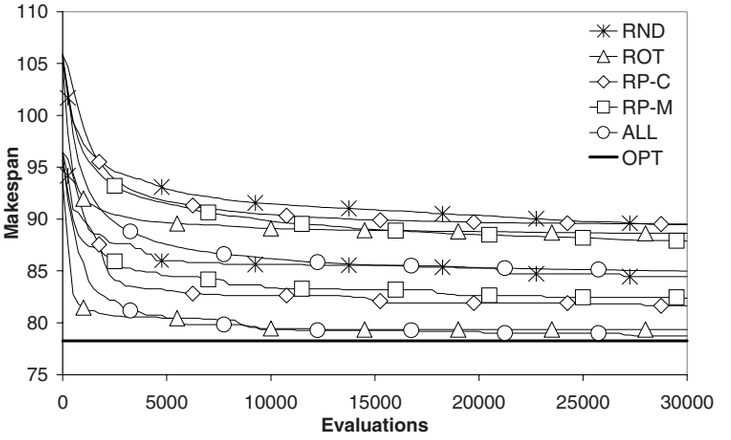


Fig. 4. Results for heuristic initial populations.

The use of more heuristic methods, such as the heuristic presented, but also during the search, is expected to improve the GA, so that a natural extension of this work must be in that sense.

5 Conclusions

A genetic algorithm has been presented for solving the assembly sequence planning problem, a much more difficult problem than other sequencing problems that have

already been tackled using similar techniques, such as TSP and JSSP, because it involves also the selection of assembly operations that will form the assembly plan from a set of alternatives. Two families of genetic operators have been used in order to search through the whole solution space. RP operators have been proposed for generate new assembly plans (with different assembly tasks) from others. On the other hand, there is the ordering of assembly tasks in the sequence to obtain a good schedule. ROT mutation operators have been used for this goal.

Although the genetic information used by these operators could seem insufficient, the combined operation of them has been quite satisfactory. The behavior of the GA algorithm has been improved by using a initial population that have been generated using a heuristic corresponding to an estimation of the time needed for executing each task and its successors in the And/Or graph of the product to be assembled.

References

1. T. Starkweather, S. McDaniel, K. Mathias, D. Whitley, C. Whitley (1991). A Comparison of Genetic Sequencing Operators. *Proceedings of the Forth Intl. Conf. on Genetic Algorithms*, ICGA-91, pp. 69-76. Morgan Kaufmann.
2. G. Syswerda (1990). Schedule Optimization Using Genetic Algorithms. In L. Davis, ed.. *The Handbook of Genetic Algorithms*, pp. 332-349. Van Nostram Reinhold.
3. Bourjault, A. (1984). *Contribution à une Approche Méthodologique de l'Assemblage Automatisé: Elaboration Automatique des Séquences Opératoires*. Thèse d'état, Université de Franche-Comté, Besançon, France.
4. De Fazio, T.L. and D.E. Whitney (1987). Simplified Generation of All Mechanical Assembly Sequences. *IEEE J. Robotics and Automat.*, Vol. 3, No. 6, pp. 640-658. Also, Corrections, Vol. 4, No. 6, pp. 705-708.
5. L.S. Homem de Mello and A.C. Sanderson. A Correct and Complete Algorithm for the Generation of Mechanical Assembly Sequences. *IEEE Trans. Robotic and Automation*.Vol 7(2), 1991, pp. 228-240.
6. B. Romney, C. Godard, M. Goldwasser, G. Ramkumar (1995). An Efficient System for Geometric Assembly Sequence Generation and Evaluation. *Proc. 1995 ASME International Computers in Engineering Conference*, pp. 699-712.
7. T. L. Calton. Advancing design-for-assembly. The next generation in assembly planning. *Proc. 1999 IEEE Int. Symp. on Assembly and Task Planning*, pp. 57-62.
8. Wilson, R.H., L. Kavraki, T. Lozano-Pérez and J.C. Latombe (1995). Two-Handed Assembly Sequencing. *International Jour. Robotic Research*. Vol. 14, pp. 335-350.
9. Homem de Mello, L.S. and S. Lee, *eds.* (1991b). *Computer-Aided Mechanical Assembly Planning*. Kluwer Academic Publishers.
10. M.H. Goldwasser and R. Motwani (1999). Complexity measures for assembly sequences. *Intern. Jounal of Computational Geometry and Applications*, 9:371-418.
11. C. Del Valle and E.F. Camacho (1996). Automatic Assembly Task Assignment for a Multirobot Environment. *Control Eng. Practice*, Vol. 4, No. 7, pp. 915-921.
12. Homem de Mello, L.S. and A.C. Sanderson (1990). And/Or Graph Representation of Assembly Plans. *IEEE Trans. Robotics Automat*. Vol. 6, No. 2, pp. 188-199.

t A go t m pp d to o m At to d t o

So a Mo a ar o Ro ra o Toro o r a

[1]D r n o d r u c ur y cno o d Co u dor , n r d d d r n d ,
n, , on c u r , du rdo c u r
D r n o d n n r c r n c , nfor co y u o c , n r d d
d Hu , n, f oro u u

a P ro y r br on P r d c on b y n o n r rc n
d f n on of n d r r for c n b b d on ry ro n ou
f ur n c or u o c y f nd of r r o
d no c b of c b d on n r n bour or n
ff c n y of n r n nu b r of o b o u on for rob of P r d c on
r n r u r o , r r n c udy of d b c n b
u d o c r r n b c f c on r

u i

r a br a o ar arr y a a o r q y a bo
o ra r bro a ar a T a o a
a o o a a r o by a a y o r r a o
o o a ra a a r a or a
o or r o ba o r ara r o ra 6
b o o a ob a r a rob r a
o
r a o a r ar or a r y b a o y ab y o a
a o a a o a or o r aro y a r a br a o
o a a o o o b a ab y
a o a a or a o y a ara r
a y or a b a o o a or a o a o

T r a o o r o ara r a r ro *C n*
r q a o T rob o a q a o by
ara r a o a o a o rob
b a rab o r ra o a o o ra r a a
o So o ba o r ara r o b a o
a r o ar a arr y a a o a a ara r
a ar ra R a T r or o oo a a a o or

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 345-352, 2003.

a o ba o a o a ara r o orr by rr ar a arr y a
o o ary a or ar a r ba o a ra o a a b a y o a ra o o a o rob a r o o o o ary o a o ay b o r or a q a b ar r o o a or r b a MM a b or abo o a o a o rob

T r a r r b a a o S o a a o ar a a o a or S o T o o o o o o a a o ro a b r b a a r a o by a o MM a y o r a ary o a o o

a aba i i a b i i i

b a aba or r o a a o a a ab ro by y oba o o o r r o a y a a a a o or a o o a o o y o a a y o a r or a o a or ro o a aba o o ara r r o a o or a ar a W r ar or a y b or a o orr o o a a b ara o o ro

- r r o or a a o r o o o a a b or a a r r or ra ar r o o
- r r o or a a a y r o o a o

T a o a r r y a o o a a or or a o a a o o a ba o ra o o o a ar y T a a a o a ab o a or o o o o o o o ay a r a o rob o a a o o y r a a a o W a a ab a aba o r o a b a r

- a o ba o ra o o T a a o a o r a r r o a a o o a aba
- o o r o T a o o o o o o a a o b r r r o o o a a o r

M u a a i i a i i

o ro a or a o ra ara r RR ra R
a a a ra a o r or a o
ara r a ra or a o ay a a
ra a o a o o or a ara r r r a
r y a a o r ra or r a o a ara r r
or a a ra M o a ara r a a a o
a aba a a ara r a by orr o M ay
or o o ar a o a a a a ar ra y a
b o ar a ara r a a o
o ar a a o a or or a a o ba o ar
bo r a b r b T ab or ar r o
a a o a or or a o ab or a a o
ab or ar a a T ab o ar bo r o
a a a a ab a a a o o o bo r
r o o o o a b ob a a bo r
ab T o ar ro r y o a or a a y o o r a
r b r o ara r o a a a o
T r or a a o a o a o o a b a o ar b o
ara r a a a a o r or a T a a o
o o ary a or
So ara r a or r a o o r a o r a a or
o o o o o ob a r r a a r b
a a y a r
r o or ara r r by a o a a a
y a roa ro o r y o or
a o a a a y a a o r o o ba o r ara r
a aba or a r a a o a a ab b o o
o a o r o ab r r a a a a a r
ro ab o or r y r r or a aba ar o
a r a o a a a a o ab o o
a o r ob a ry r r To a a
a aba ro ra a r r a a r ra
a ara r ara r T a a o a or r b abo a
o r o r a b ra a or ob a
a a o r aro T a a b a
r ra a r or ar o a a b ar or
a a o ba o ara r

a i i a i u G MM

o o ary a or r o a a a a
o o ra o by a o ra or o o o Ty a ra or a o
o ra or a o ra o a a o o a ro o r a o a

r o b a o b a y o o *n* o a a o o
T o o q a y a a by a o a o T or o
oo ra ra o ro b r r or a a a ar o
o r o o ary a or ar ar o
o o r o o o ar a a o ra
o a a r y a o a o ay b r q r o a o a o
rob r or a a o rab o 6
r by b a o r o o y r o r o ar b
a o a o o ra
or a b a or o r r o T
a or r or a o q a or ora b o a
a o a q o a r ro
or r o b ab o o y o r ara r
a a o a o a a y a r by a or
W by W or a a or
b a r orr o ara r T or r r
ro o o o r o o o by MM o a
a a o ra o q a o T MM a
a o a o o a a o ra o a ra o W a
ob a r r or r o o o

4. a o ba o o y o ra o a y
r o o o a o a ro o

or a roa or orr o o a a a a by a
ara r or ra ro ra r o o a o a
ara r or ra ro ra o r o o a o T
ra a o a o o o o ara r by r W or

a e r u of P d no b d on dd on for r r r c d of C r c f r fro P od nd r r r c d fro C r c d y r ou o P od

l t n	**m m ng t t t**
4	

l t n	**l t n m n**	**n t t**	**p t**	**mb t t**
			6	
			6	
			6	
4			6	

T ob a o o a a o r or a abo
r r o o a o a o or a or a y
o ra o o r b o o o a a a o r or a abo
r ara r ar ar Tab
To ra a o r o o ar o a y r a a
a a b a b r o *n mm n m n n* 6
a T a r r r
b o o or a o o a o r ar ba o
a r a o y o ar ar b a a

4. a o ba o ra a y r o o a o

T ob a o o a a a o r or a abo
r r o o a o a o T o r b o o ar ar
Tab

a e r u of P d no b d on C r c d y r ou o P
od

l t n	**m m ng t t t**
4	

l t n	**l t n m n**	**n t t**	**p t**	**mb t t**
			6	
			6	
4	6		6	

T a b r o o o o o b r
o o ar 6 a T r o
ara r a o r o o a o y ar a r o o

4. a o ba o ra o r o o a o

T ob a o o a a a o r or a abo
r r o o o a o Tab ar b o r o o

T a b o o ar a
S o o ar a o r o o o o ar a
r o o a o r ar a o o o a ara r ar
y a o o

a e r u of P d no b d on C r c no r ou o P od

l t n	**m m ng t t t**
4	

l t n	**l t n m n**	**n t t**	**p t**	**mb t t**
			6	
		6	6	
4				

4.4 o o r o

T ob a 6 o o a a o r or a abo r r 6 o o o a o Tab ar b o r o o

a e r u for P od on r d c on

l t n	**m m ng t t t**
4	

l t n	**l t n m n**	**n t t**	**p t**	**mb t t**
		6		
		6		
			6	
4			6	6

T a b a o o ar a S o o ar a o r o o 6 ar a r o o o o ar o o o ar y by a o o a a r ar o a y o o

i u i

T a r a r a a o o r o a o ba o ra o o o r W r o r a o

a a o r o or ara r a a b
ra ro ra or a a o T a o o o o
a a a o ra o a a or ba o ar
bo r o a r a o a r a or a ra
o o r r by a or a a b o r ara r
or or r o a a r ara r o a r a
a a o ra o aro or a o ra r o o ba o
r ara r a a o b r a y o ara r
T ara r a a o b or o r o a a o
ra o o T a a o r ar a o ar o r
a roa T ra o o a o o a o o o o r
ara r r o o a b a a o a
b a o a a roa o o a b
o r a r a o a r or o b r o ob a a o
a or y T ara r a a ab y o
a a a o ra or a y b r a
T r or o o y r rab a o o o
a a a o r or a T a a ab y o a r o
o o ba o r ara r a r b y o
oo a a a o a ar ar ara r ara r by a
a or a ro r a or a ar ar b a ay r o o r
ar a a o o a ay a r a ara r
a o r o o a a a o a or o ar a o
o o ara r by o r a or ra or ar

i

or b o a a o a a o a a o a a a **ll**
r a r a a o o y or **lt** r ro ar ar a o o y T
a a a a o r a b a o o o o a
t . T a or a b a a b
a
g t . T a or a b a a y a
b a a y
l t . T a or a b a b b
a y
l g t . T a or a b a a y b
b
W a r o o r a b ***ssi ic i***
si ivi y ra o b a a o a a a
si ivi y ra o b a a o a a

$$C = \frac{P+}{P+ \quad + \quad P+} \qquad I = \frac{P}{P+} \qquad P \quad CI = \frac{}{+ \quad P}$$

R

C b n H C ubb H C P r u r n n " of y c
bo on of r f br on ou r no ", C rd o , o ,
,
P r n P od fr d n " bo c co c on n ro y r f br on",
ro , o ,
o o N, o , no n r d P bnor nd r n
n nd ou d o c ro y r f br on H r ,

r D, o c r N, n H, u , n bur , nb r n r d P
dur on do no r d c r f br on f r or c c ur ry n o o y,

n , o H, yry , P uj , o o , n n n P,
r n n , Hu ur H r d co y nd corr on ro r of
n r dyn c b for on n ou on of ro y r f br on
C rcu on,
Hn o , r P, ur royd D, uo , y n , C ,
n y of c rd c r y r c d n od of ro y r f br on
H r
o b C, Nurnb r r , Ndr , c r c , r nn r , rc , c C, od
of n on of ro y r f br on n n y of on n ou y occurr n
od u n d Ho r on or n P C ,
or o a o
o db r r , r N, , H u dorff , no PC , r , u ,
oody , P n C , n y H P y o n , P y o oo , nd P y on
Co on n of N rc ourc for Co P y o o c n *C r n*
C rcu on c ron c P
http://circ.ahajournals.org/cgi/content/full/101/23/e215, un
d oro, D , C , r *A MM: O M P e*
A e c d orn d d P r o, nc ,
c , H , c f , H P o u on ry Co u on co n on
ory nd curr n r n on o Co , o , No , br ,
y on c un r y ob n d b f db
o , o , rn nd , , D , Pr o, C P r r C r c r on of
P ro y r br on , , Co o, y, un
o , o , oro, , D , rn nd P ro y r br on
u o c D no or b d on no br n C , ,
Co o, y, un
r n o , n o u on ry r y for ob n on nd r o c n
n y r n on o u on ry Co u on, o , No , ,
r n nd r nb , "*F e Me e e* "
r n c on on o u on ry Co u on, o , No ,
C n P , ur y or P r nfor cn co ,
foud, *"A f P e e e Me* ", n
Conf r nc on n c or , or n uff n, n o, C ,
D j r, P y r, *P rm R n n r* , Pr n c
H n ood C ff
oody, , o db r r, cC nn n, P ryn, "Pr d c n n of
P ro y r br on Co u r n C rd o o y C n ", Co u r
n C rd o o y , o rd , ,

Optimizing supply strategies in the Spanish Electrical Market

Enrique A. de la Cal Marín and Luciano Sánchez Ramos

Universidad de Oviedo, Dpto. de Informática, Laboratorio de Metrología y Modelos,
Campus de Viesques, 33203 Gijøn, España,
delacal@etsiig.uniovi.es
luciano@ccia.uniovi.es

Abstract. The price of electrical energy in Spain has not been regulated by the government since 1998, but determined by the supply from the generators in a competitive market, the so-called "electrical pool". In this work, we present a genetic algorithm-based method that allow us to simulate Cournot-like equilibrium situations in the Spanish pool. A perfect oligipolistic behavior of the agents will be assumed. The output of the algorithm being proposed here contains an approximation to the optimal supply curves of the competing agents.

1 Introduction

The relationship between the cost of production and the selling price of electrical energy is not direct. Production cost determines price in a regulated market, such as the one that existed in Spain before 1998 and continues to exist in other European Community countries. This is not so in a competitive market. Before 1998, prices in Spain were fixed by a public agency that was also in charge of elaborating a list of the power plants that should connect at any given time. This list was calculated by means of numeric optimization algorithms, which minimized the global cost of the production necessary to cover domestic demand.

In the modern model, based on free competition among the different companies [1], the Market Operator (MO), a neutral agent appointed by the State to regulate competition, calculates the energy prices for every hour, starting from the supply of the generators and the demand of the consumers, this process is called "casation procedure".

The procedure by which production is planned is based on the principle that "the cheapest power plants connect first". In this case, however, "cheapest" does not mean "low cost", but "low selling price", because each agent is free to choose the price it wants to charge for its power. It is interesting to note that the law stipulates that all power plants are to receive the same payment for each MW of energy sold, as occurred in the non-competitive model, and not the payment they asked for in their strategy. The second principle of the competitive market is "the most costly power plant connected marks the price".

In previous works [2], we designed a genetic tool able to estimate past offers from the agents in the pool from publicily available data: hourly prices and

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 353-360, 2003.

amount of energy consumed. This last model was a useful analysis tool in certain situations: it allowed us to estimate the change in the price of the electrical energy under small deviations of the supply curves of one agent. The simulation procedure was as follows:

1. The supply curves of all agents are estimated by means of the genetic tool.
2. One of the supplies is slightly modified. For example, the selling price of a group of generators is lowered.
3. A new simulation is carried with the modified suppy curve and the original demand. Supply curves of the remaining agents remained untouched. In other words, it was assumed that the agents *did not react to the change*, changing their prices to maintain their market share.

In this work we intend to generalize the previous tool by dropping this last assumption. In other words: what would happen if the agents are allowed to react? We would be confronted with a game theory problem, which we will solve by means of cooperative-fitness based genetic algorithms.

1.1 Summary

The remainder of this paper is arranged as follows: In Section 2, the problem to be solved is explained. The "casation process" in term of Games Theory is described in Section 3. Our methodology is described in Section 4. In Section 5, a simple problem is solved to illustrate the use of the method proposed here and regression- and Game theory-based methods, applied to a semi-synthetic problem, are compared.

2 Definition of the problem

2.1 Electrical pool as a game theory problem

Given that preliminary data are insufficient to carry out a statistical analysis, it is necessary to make conjectures regarding the results. This work assumes that the agents are intelligent and that the market is fair, such that the unit profits (euros/MW) are approximately the same for all the competitors.

With these hypotheses, if we know the cost of production of the agents (and we can estimate that, using data prior to 1998), it is possible to simplify market operation and abstract it to a game, which can be explained as follows. Let us assume that a certain amount of energy is to be bought from several generators. None of them is capable of supplying the total amount and the amount supplied by all of them exceeds the needs.

Each player (one of the generators) gives a referee (the MO) a closed envelope with its sales strategy. It consists of a pair "quantity supplied - price demanded per unit". The referee opens the envelopes, arranges the strategies and chooses the cheapest ones until demand is covered. Each player selected is then paid for the amount it sells at the price of the most expensive strategy that was accepted.

Each player receives the difference between the price paid and their unit cost, multiplied by the energy units sold.

The actual number of players is several hundred (one player per electrical power plant). To simplify calculations, we group the price-quantity pairs of all the power plants belonging to the same company into a single total quantity produced-unit price curve. In this way, we reduce several hundred strategies to four aggregate supply curves (there are four large electrical companies in Spain). The same is done with costs: each of the four participants in the simplified game will have a curve that relates the negotiated MW with their production cost. The mechanism of this new game is a bit more complex: each player gives the referee an aggregate supply curve. The referee adds up all the curves and intersects the results with a demand curve. The cross point determines the market price. Given the price and the supply curves furnished by the agents, the revenue of each player is calculated.

Finally, the net profit of each player is calculated using the difference between the income received and the value of its cost curve at the point corresponding to the amount negotiated.

3 Proposed methodology

In this work, we want to estimate the optimal supply generation curves of agents competing in the electrical pool from real data taken from an hourly database provided by the MO. We will define a nonlinear parametrization of these curves and use a coevolutionary genetic algorithm to find an equilibrium point generalizing Cournot equlibrium to our own definition of supply curve. Briefly, the genetic algorithm works as follows: first, we define as many populations of strategies as players. To score a strategy, we will simulate a game, making this strategy to compete with a selection of strategies taken from the remaing players.

The algorithm studied in this work serves us to obtain the optimal (in Cournot terms) supply curves of the companies competing in the market using demand curves of several previous markets. The input data are the demand curves and the costs of generation of all competing firms. The outputs –the supply curves– represent market strategies.

3.1 Definition of a suppy curve

Strategic Planning Departments take into consideration the day of the week, the hour of the day, the season, the weather forecast (rain, temperature) and some other indicators before posting prices to the Market Operator. Following our own experience, three features should be considered: the hour (which is related to the amount of energy negotiated, depending on labor hours and daylight), the day of the week (the dependence between labor hours and demand changes on weekends and holidays) and the season (electrical cooling or heating, affects both previous dependencies).

Given this information, we decided to stay in an intermediate position between (a) assuming that the supply curve is always the same for each agent, and (b) assuming a different curve for every market. Since (a) is too imprecise and (b) is intractable, in this work we will allow each agent to select its curve from a restricted set of choices, depending on the values of the features mentioned before. In other words, a strategy comprises:

- a rule-based classification system, that produces a segmentation of the market points into a certain number of classes depending on hour, day of the week and type of day, and
- as many supply curves as market segments.

That is, each individual is a set of rules whose antecedents are assertions with regard to market characteristics and whose consequents are the supply curves that the player can use. We shall call these consequents "prototype strategies".

The simplest representation of a prototype strategy is a straight line. Linear models can approximate the behavior of a competitive electrical market in the neighborhood of its equilibrium point. Unfortunately, in spite of this kind of simplification, which is valid for studying the response of the market under small changes, it is not accurate enough to estimate complete supply curves of the agents, which are highly non linear. We have decided to use piecewise linear supply curves instead (see Figure 1.) Their number of segments will be a compromise between the accuracy of the model and the amount of available data (three segments in most of the experiments in this paper.)

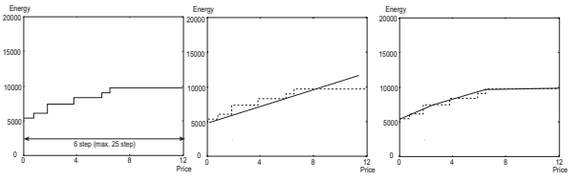


Fig. 1. Actual (left), linear (center) and polygonal supply curves (right). Representation by a polygonal line is closer to reality than the linear supply and does not depend on an excessive number of parameters.

3.2 Genetic representation

Each individual in the coevolutionary approach [4], [6] codifies a possible set of strategies (i.e., a fuzzy-rule-based classifier system and a set of prototype strategies) of one of the agents; we will keep as many populations of individuals as agents exist. Fitness is not assigned to an individual but to a combination of individuals extracted from all populations [7] [8].

An individual in the coevolutionary approach will be codified with a chain of numbers. This chain comprises two real numbers to define every segment in a prototype, plus a list containing the numerical parameters on which the linguistic terms in the antecedents of the classifier depends.

3.3 Genetic operators

Individuals in the coevolutionary approach are represented by chains of real numbers, thus there is no need to define custom genetic operators. However, the relative sizes of the subchain codifying the consequents and the subchain codifying the list of parameters of the classifier are very different. We have opted to let only one of these parts be modified in every genetic operation, thus we can manually balance the evolution of both and speed up the evolution of the classifier part. This is the only difference between our operators and the standard versions of uniform arithmetic crossover and mutation [9][10].

When two individuals are to be crossed, a coin is tossed to decide whether we select (a) the subchain codifying the classifier definition or (b) one subchain that codifies the definition of one of the prototypes. The selected subchains are recombined by means of standard arithmetic crossover, but the remaining part of the individual remains untouched.

The mutation operator is defined as the crossing of an individual with another one, generated at random.

3.4 Fitness function

Each firm has as an objective the maximization of its own profit, assuming that the unitary benefits are all the same. This decision implies that we need to rank strategies according to two different criteria. Let us compare two strategies f_1 and f_2 of the same firm: after playing games with either f_1 and f_2 and all strategies of the remaining players, an strategy f_1 is better than another one f_2 (a) the best aggregated profit of the remaining players against f_1 is lower than their benefit if f_2 is used (min-max strategy,) and (b) f_1 produces similar unitary profits for all players. "Unitary profit" is defined as the difference between cost and income, divided by the number of MWs sold. The first goal measures the benefits of a strategy in the worst case, and the second one measures the degree of fulfillment of the restriction "all unitary profits are the same." We will use the profit of the strategy to quantify the first objective, and the mean of the variances of the unitary profits to quantify the second one.

Different methods exist for mapping multi-objective fitness into scalar fitness [5]. We have studied the weighted average of values (a) and (b), but, according to our experiments ,[2], there is a significant improvement if we use a multi-objective approach instead [3].

4 Numerical Results

A simple problem is presented here for the sake of illustrating the basic aspects of the proposed methodology. This example models 10 repetitions of a game in which there are four players with known cost curves that always use the same supply curve. Supplies are straight lines, each depending on two parameters.

Table 1. Equilibrium Market Point Calculated Analitically

Market	0	1	2	3	4	5	6	7	8	9
price	3,3	4,1	4,9	5,71	6,5	7,4	8,2	9,1	10,0	10,8
energy	1720,7	1925,7	2115,2	2292,3	2459,3	2617,6	2768,5	2912,9	3051,7	3185,4

The inputs for this problem are:

1. The cost functions. q is the quantity of energy produced, C_0 to C_3 are the prices demanded: $C_i(q) = (4+i)e - 6q^3$
2. The market scenario, a series of 10 demand functions (D_m) with the same elasticity (i.e., the same steepness): $D_m(p) = -1000p + (5000 + 1000m)$, for m in $0 \ldots 9$
3. The set of Cournot market equilibrium points in table 1, (price$_i$, quantity$_i$) calculated analitically.

Table 2. Equilibrium Curves calculated by Analytical Method for 10 Market Points

Market	Firm0	Firm1	Firm2	Firm3
0	150,43p	134,58p	115,46p	124,25p
1	135,57p	121,27p	103,92p	111,87p
2	124,26p	111,13p	95,16p	102,46p
3	115,29p	103,11p	88,22p	95,00p
4	107,96p	96,55p	82,56p	88,92p
5	101,84p	91,06p	77,83p	83,84p
6	96,62p	86,39p	73,80p	79,51p
7	92,10p	82,36p	70,32p	75,77p
8	88,15p	78,82p	67,28p	72,50p
9	84,66p	75,69p	64,59p	69,61p

Now, estimated equilibrium individual strategies were obtained after running our equilibrium coevolutionary algorithm with four populations with the size of 1000, Cournot multicriteria fitness, 400 generations, tournament selection (size 4) and linear descending crossover probability, from 100% to 0%. The output of our equilibrium method is:

$$q'_0(p) = 67.1636p + 226.987; q'_1(p) = 56.4915p + 234.217$$
$$q'_2(p) = 45.5331p + 272.999; q'_3(p) = 41.5025p + 258.445$$

Every strategy q'$_i$ will be applied to each market m (demand curve D$_m$).

Estimated equilibrium aggregated supply curve for these market points is in Table 1. We can see, that our estimated aggregated supply curve is a good approximation to the non-colinear cloud of analytical market points. The mean error is 2.65361% by market and firm respect to energy dimension. The mean percentage error of estimated individual quantities for each market was 5.93% and the mean percentage error of estimated energy share in each market was 1.11%.

4.1 Semi-synthetic problem

This section describes the application of our method to a semi-synthetic problem componed by 40 market points. This problem was designed to reproduce current scenarios in the Spanish electrical market, while being originated by theoretical data, thus we can assess our results. The methods proposed here will be called "Equilibrium Coevolutionary Genetic Model" (ECGM) and "Regression Coevolutionary Genetic Model" (RCGM). The first method is the one being proposed in this paper and tries to obtain the optimal oligopolistic solution, while the second one was proposed in [2], and obtains the actual supply curves being used in the market.

Individual strategies extracted by RCGM from 40 market points (price, energy, energy level, type of day, temperature) score 100% linguistic matching respect theoretical individual strategies used to generate market points. The mean percentual error of estimated individual quantities for each market was 2.92% and the mean error of estimated energy share in each market was 0.69%. Results of competition of the strategies resulting from applying RCGM and ECGM to semi-synthetic problem respectively are in table 3. It can be seen that ECGM pool obtains 159.852% more profit than RCGM pool generating 16.716% less energy, thus there is still room to obtain more benefits if supply curves are improved. Notice that the opposite result (better results than in the oligopolistic equilibrium) would have meant that two or more of the agents have signed an illegal agreement.

5 Concluding Remarks

We have experimentally shown that a coevolutionary genetic model can achieve the Cournot equilibrium for a set of electrical markets. The tool proposed here serves improve the profit of an electrical company by adjusting it to the supply in terms of an oligopolistic marker. Moreover, this tool may also serve to the Market Operator, because the estimation of the theoretical maximum profit and its comparison to the actual situation can be used to detect illegal agreements between generators.

Table 3. Energy and Profit Comparison for RCGM and ECGM Resulting Pools

	Firm 0	Firm 1	Firm 2	Firm 3
ECGM Pool Profit	198143	197669	195047	252652
RCGM Pool Profit	125191	156961	85585.2	135626
% Profit Variation	-36.818	-20.5943	-56.1208	-46.3189
% Global Profit Variation	-159.852			
ECGM Pool Energy	140347	148200	145383	193681
RCGM Pool Energy	98210.3	196062	141886	226314
% Energy Variation	-30.0231	+32.2956	-2.40524	+16.8487
% Global Energy Variation	+16.716			

Acknowledgements

This work was funded by Spanish M. of Science and Technology and by FEDER funds, under the grant TIC-04036-C05-0 and Banco Herrero Research G. 2002.

References

1. Ministerio de Industria y Energía. Protocolo para el establecimiento de una nueva regulación del sistema electico nacional. Technical report, Ministerio de Industria y Energía, 1996.
2. E.A. de la Cal and L. Sánchez. Generative offers estimation using coevolutive genetic algorithms in tha spanish electrical market. *Applied Intelligence (Admitted for publication)*, 2003.
3. C. M. Fonseca and P. J. Fleming. Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms—Part II: A Application Example. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 28(1):38–47, 1998.
4. W. D. Hillis. Co-evolving parasites improves simulated evolution as an optimization technique. *Artificial Life II*, pages 41–47, 1991.
5. Y.-J. Lai and C.-L. Hwang. *Fuzzy Multiple Objetive Decision Making: Methods and Applications*. Springer-Verlag, New York, 1996.
6. K.A. De Jong M.A. Potterm and J.J. Grefenstette. A coevolutionary approach to learning sequential decision rules. *ICGA95*, 1995.
7. J. Paredis. Coevolutionary computation. *Artificial Life*, 2(4):355–375, 1995.
8. C.D. Rosin and R.K. Belew. New methods for competitive coevolution. *Evolutionary Computation*, 5(1):1–29, 1997.
9. W. M. Spears and K. A. De Jong. On the virtues of parameterized uniform crossover. In Rick Belew and Lashon Booker, editors, *Proc. 4th Int. Conf. on GA*, pages 230–236, San Mateo, CA, 1991. Morgan Kaufman.
10. G. Syswerda. Uniform crossover in genetic algorithms. In Rick Belew and Lashon Booker, editors, *Proc 3th Int Conf on GA*, pages 2–9, San Mateo, CA, 1989. Morgan Kaufman.

Improving the Efficiency of Multiple Sequence Alignment by Genetic Algorithms*

Juan Seijas[1], Carmen Morató[2], Diego Andina[1], and Antonio Vega-Corona[1]

[1] Departamento de SSR, E.T.S.I. Telecomunicación
Universidad Politécnica de Madrid
Ciudad Universitaria s/n. 28040 Madrid. Spain
juan.seijas@sener.es
[2] Departamento de Matemática Aplicada
E.T.S.I. Agrónomos
Universidad Politécnica de Madrid
Ciudad Universitaria s/n. 28040 Madrid. Spain
mmorato@mat.etsia.upm.es

Abstract. Multiple alignments of biological nucleic acid sequences are one of the most commonly used techniques in sequence analysis. These techniques demand a big computational load. We present a Genetic Algorithms (GA) that optimizes an objective function that is a measure of alignment quality (distance). Each individual in the population represents (in an efficient way) some underlying operations on the sequences and they evolve, by means of natural selection, to better populations where they obtain better alignment of the sequences. The improvement of the effectiveness is obtained by an elitism operator specially designed and by initial bias given to the population by the background knowledge of the user. Our GA presents some characteristics as robustness, convergence to solution, extraordinary capability of generalization and a easiness of being coded for parallel processing architectures, that make our GA very suitable for multiple molecular biology sequences analysis.

1 Introduction

The best way to discover relationships among genes is through analysis of their nucleic acids and proteins sequences. They have to be properly represented as extended strings from predetermined alphabets. Those alphabets use four "characters" {a, c, g, t} when they are expressing the nucleic acids. The panoply of algorithms for multiple sequence alignment has been growing tremendously. In last decades, new methods have been developed: Some are exhaustive methods, based on dynamic programming, heuristics like MSA, and modifications like MAP [1]; for obtaining the alignment of many sequences there are programs, with a sub-optimal behaviour [2], implementing the progressive alignment technique.

* This research has been supported by the Spanish National Research Institution "Comisión Interministerial de Ciencia y Tecnología-CICYT", Project TIC2002-03519.

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 361–368, 2003.

All these methods require a heavy computational load, and other optimization alternatives like the Markov modelling of the sequences [3] were proposed. Still demanding however a big computational power when dealing with many sequences: The computational load depends on $O(k * n^2)$, where "k" represents the number of sequences and "n" is the length of them (assuming they all have the same one).

1.1 Genetic Algorithms for Sequence Analysis

Multiple sequence analysis requires finding alignments with the minimum number of changes in the sequence structure. GA are capable of finding the optimal alignment or sub-optimal alignments as well as those found by other procedures classical in biology. Starting from our previous research, where two sequences comparison algorithm [4] and three sequences comparison algorithm [5] where developed, we have improved those algorithms, introducing new operators and functions capable of optimizing the alignment of multiple sequences.

2 Problem Formulation

In our previous research we have presented a very efficient GA for multiple sequence comparison [6]. However, the choice of the alignment structure is a big issue in sequence comparison, especially when the sequences to be aligned have a loose coupling among them. Taking advantage from GA robustness, we have recently added several modifications in the GA architecture in order to improve GA generalization and flexibility when dealing with very unlike sequences.

2.1 Genetic Algorithm Description

For biological sequences comparison and best alignment identification we designed a GA derived from the simple (and multipurpose) ones presented by [7]. The GA starts with a random population; the individuals of the population represent some operations on the biological sequences. After carrying out those operations, the alignment of the sequences is computed for each individual by the objective function. The genetic operators are applied to the population in order to create a new population, this new population is evaluated by the objective function and the loop re-started until some of the break criteria are fulfilled. Reproduction, crossover and mutation operators are applied to consecutive populations in order to evolve to a new population, where the performances (measured by the objective function) of their individuals (one of them at least) are better.

2.2 Elitism Operator

In general, GA performances are improved when outstanding individuals are automatically passed to next generation. However, this strategy tends to premature convergence in the optimization. Several improvements to elitism have

been presented [8], but they do not pay attention to the nature of the problem under optimization. For a GA working in the deterministic application, an elite of one individual could be enough to improve performances without premature convergence. But this elitism strategy is a bad approach when working in a stochastic application like ours: the same high-fitness individual at iteration **"t"** may present bad performances at iteration **"t+1"** when its chromosomes are demanding improper character replacements. To avoid it, the first step is to increase the number of individuals in the elite. However, the higher this number the quicker population looses diversity. In our simulations, 30% of individuals at the elite configured the maximum allowable. We implemented the following restriction: After iteration **"t"**, the 30% of the individuals were pre-selected as candidates to the elite, but before being passed to next generation, they were measured again by the objective function. Only those individuals who are now in the top 10% of the rank are selected, and the rest are disregarded for the elitism operation. In most of the iterations, the best individual was not selected, especially at the beginning. The performances of this elitism operator were described at [9].

2.3 Objective Function

Evaluation of the alignments is carried out using an Objective Function. For more M sequences, it seems reasonable to define the objective function f_M from the sum of all the partial $f_{i,j.}$ fitness function, where i,j sweep all M possible pairs. This definition of f_M causes a bad behavior of the GA because it can force solutions with a lot of blank characters, and this is not easily avoided by reducing the weight$_{blank}$. So, for M sequences, f has three terms: The first term, f$_{match1}$, is obtained by computing the matches between two sequences, one fix sequence (no matter which one is selected, let us consider sequence_1 and the other sequences. The second term contains the matches between all the possible pairs of sequences where sequence_1 is not present. The third term computes the entire blank simultaneous among all the sequences.

2.4 Individuals Codification

For the sake of clarity let us consider only three sequences, M=3. Only M-1 sequences are going to be rotated in relation to one fix sequence, so each individual, x$_{i,}$ has to represent a certain number of rotations and a certain number of blank insertions on sequence 2 and 3. The "genome" of individual "i" is [R$_{i2}$ B$_{i2}$ R$_{i3}$ B$_{i3}$]. The genome of individual "i" is contained in row "i" of the population matrix. The genome has two parts: one related to the first sequence to be manipulated [R$_{i2}$ B$_{i2}$] and another related to the second sequence to be manipulated [R$_{i3}$ B$_{i3}$], each part is codified in two "chromosomes": R$_{ij}$ represents the number of rotations requested by the individual "i" to sequence "j" (j=2 or 3). B$_{ij}$ is also an integer number, but when it is translated into its binary format it is a 2* l$_{seq}$string of 1's and 0's (0 ¡ B$_{ij}$ ¡ 2^{2*lseq} –1), now each pair of bits placed at position "2k-1" and "2k" means: '00': No change at position

"k" of the sequence. '10': Blank insertion at position "k" of the sequence. '01': Character elimination at position "k" of the sequence. '11': Character replacement at position "k" of the sequence. With this extended codification scheme, all the operations on the biological sequences are provided, giving to the GA the capability of finding better alignments because of the enhanced search space. Doing so, the whole population of N individuals is codified in an N_x_4 matrix, N_x_2(M-1) matrix, for M sequences.

$$Population = \begin{bmatrix} R_{12} & B_{12} & R_{13} & B_{13} \\ R_{22} & B_{22} & R_{23} & B_{23} \\ \vdots & \vdots & \vdots & \vdots \\ R_{N2} & B_{N2} & R_{N3} & B_{N3} \end{bmatrix}$$

The number of rotations applied to the sequence "j" by individual "i" is codified in a plain integer number, namely R_{ij}. That number means R (rotations) shifts to the left of the characters of the sequence, and it varies between 0 and l_{seq}-1 (l_{seq} is the length in characters of the sequence).

2.5 Parameter Selection

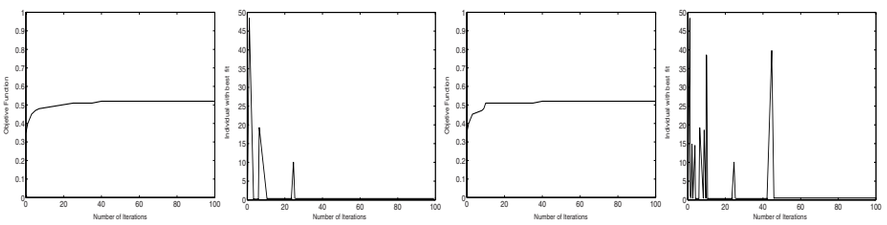
The internal parameters are: Number of individuals "N", the probability of crossover "P_c" and the probability of mutation "P_m". All the simulations run have demonstrated the robustness of the algorithm in the sense of low sensitivity of the algorithm's performance to the internal parameter values. However, good performances are obtained with $N \leqslant 50$, (more individual increase the computing load without reducing the number of algorithm's iterations before convergence) and with $P_c = 0.3$ and $P_m = 0.03$, with these values the iterations can be stopped at the 100^{th} generation because no better solution is usually found later on.

3 Improvements

The improvements that we have implemented are: Objective Function Scaling, Biased Initialization and Character Substitution Avoidance. They offer to the researcher the possibility of driving the convergence process when the disparity among the sequences forces the GA to stagnate.

3.1 Objective Function Scaling

We use a cubic scaling $fit = \left(\frac{f}{f_{max}} * 100\right)^3$ for the objective function to keep adequate levels of competition among the individuals of the population. It may happen, at advanced stages of the convergence process, that the average scoring for the whole population is very close to the best individual score. At that moment, almost everybody in the population has already the same probability to pass to the next population; in this situation the survival of the best individual is threatened by the middling individuals. The best individual selected by the elitism operator is "usually" from the first places of the population ranking (see Fig.1(b)), and, occasionally, a new individual, coming from the last places in the



(a) (b) (c) (d)

Fig. 1. (a) Objective function vs. iterations, (b) individual rank before scaling objective function (c) Objective function vs. iterations and (d) individual rank after scaling objective function

rank, becomes the best individual (it is selected for the next population by the elitism operator). A symptom of population stagnation is the continuous selection of the first individual by the elitism operator. The change of the objective function to a cubic scaling enhances the differences among individuals and the population opens itself to a wide search space where new best individuals appear from not only the first places of the rank. The effect of this improvement is depicted in Fig.1(a)-1(b) and 1(c)-1(d). This prevents the GA to converge very quickly in a local maximum.

3.2 Biased Initialization

In our previous GA, the initial population were randomly generated with a uniform distribution. The population is codified in binary and it represents a wide search region because all the possible solutions have the same probability to appear in it. In other words, for a big number of individuals, there are almost the same number of "1" bits than "0" bits. In our coding scheme, each pair of bits in B_{ij} represents a special operation on the sequence, and all the allowed operations have the same probability in the initial population. For many cases of sequence alignment, it is convenient to search the best solution favoring one type of operations instead of other. This can be done by assigning a probability (P_{xxx}) to each operation: P_{ins} to character insertion, P_{sus} to character substitution and P_{eli} to character elimination. The improvement consists in passing a filter to the individuals of the initial population in order to accommodate the operations represented by their individuals to the selected P_{ins}, P_{sus} and $P_{eli.}$During the convergence process, if a mutation in one individual creates an operation that initially was restricted by a low P_{xxx} , that operation has to produce a significant advantage to the individual, and if not, it will be suppressed by the evolution process.

Example of Biased Initialization In the next examples it can be appreciated how this initial bias affects the first iterations of the GA, forcing the search in

regions where no many character operations are allowed. The three sequences are:

```
S1:asvltqppsvsgapgqrvtisctgsssnigaghnvkwyqqlpgtapklli
S2:qsvltqppsasgtpgqrvtiscsgtssnigsstvnwyqqlpgmapklliy
S3:evqlvqsgggvvqpgrslrlscsssgfifssyamywvrqapgkglewvai
```

Note: they have been reduced in length for clarity. When there is no initial bias, the best alignment after 2 iterations is:

```
Op_S1,R=3:IIEIS.IESIISSEISE.S.IEESSIE.....................
S1:__t_wp_sp__mhP_gRVgI_Cgkr_nigaghnVkWYQQLPGtAPKLL__
S2: qsvltqppsasgtPgqRVtIsCsgtSSnigsstVnWYQQLPGmAPKLLiy
S3: vqsgggvvqpgrlrlscsss_gfifSSyamwvrqpgkglewvaevqLvvv
Op_S3,R=4:...........E.......I.........E...E........E......
Score = 0.6378
```

The row Op_S1 represents the insertions (I), Eliminations (E) or Substitutions (S) carried out to the character placed at the same position. The dot (·) means no operation. R represents the number of rotations in the sequence. When there is no initial bias, the best alignment after 27 iterations is:

```
Op_S1,R=3:IIEIS.IESIISSEISE.S.IEESSIE.....................
S1:__t_wp_sp__mhP_gRVgI_Cgkr_nigaghnVkWYQQLPGtAPKLL__
S2:__t_vp_si__ykP_iRVeI_Cgks_nigaghnVkWYQQLPGtAPKLL__
S3:vqsgggvvqpgrlrlscsss_gfifSSyamwvrqpgkglewvaevqLvvv
Op_S3,R=4:...........E.......I.........E...E........E......
Score = 0.6375
```

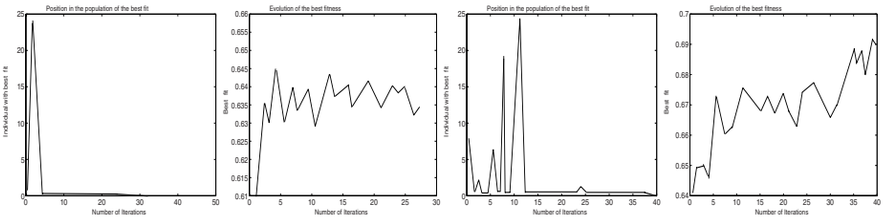
And the convergence process is represented in Fig.2(a) and Fig.2(b), When the initial bias is set to : $P_{ins} = P_{sus} = P_{eli} = 0.01$, the best alignment after 2 iterations is:

```
Op_S1,R=3:.....................I.....................
S1 aSVLTQPPSvSGaPGQRVTISCtG_SSSnigaghNvkwyqqLpgtApkll
S2 qSVLTQPPSaSGtPGQRVTISCsGtSSnigsstVNwyqqlpgmapklliy
S3 vqsgggvvqpgrslrlscss_gfifSSyamywVrqapgkgLewvAievq
Op_S3,R=4:.....................I.....................
Score = 0.64
```

the best alignment after 31 iterations is:

```
Op_S1,R=0:.....................EI..S...I.......EE...I.E.....
S1 aSVLTQPPSvSGaPGQRVTIc_tGkSSN_igaGhNvkyQLPG_tPKLLaa
S2 qSVLTQPPSaSGtPGQRVTIScsGtSSNigsstvNwyqQLPGmaPKLLiy
S3 sggfvQPgrs_lrlcvsVifSsyamyw_vqapGk_glw_v_aievqLqss
Op_S3,R=6:.E.S......I...ESESE........I.E....I..EI.I......E.E
Score = 0.6902
```

And the convergence process is represented in the Fig.2(c) and Fig.2(d).



(a) (b) (c) (d)

Fig. 2. (a) Individual rank at each iteration, (b) Objective function vs. iterations, (c) Individual rank at each iteration and (d) Objective function vs iterations

3.3 Substitution Avoidance

When a character substitution is performed in a sequence, the scoring of the objective function may oscillate from one iteration to another, due to the fact that the replacement is randomly selected at each iteration and that changes in the selected character produce different scorings. If the best individual of the population has several substitutions in its genome, it can affect its position in the rank and even force it to disappear. For this reason we let the GA to reduce the number of substitutions contained in the genome of each individual. This limitation is very common in Nature, because it means a real mutation that is not frequent in living beings. Note: When substitutions are forbidden, the behavior of the application returns to deterministic, and then, the special elitism operator (paragraph 2.2) is no longer required.

Example of Substitution Avoidance In the next examples it can be appreciated how this Substitution Avoidance accelerates the convergence process. The three sequences are,

```
S1:atgcatatgcataattaacttgcaaatcgaatcaaaagctattattctga
S2:acatatattgcattaaaactggcacatgaaataatgttacgatatacgca
S3:aattgagcggcataaccttattcacccaacggattgagcgcaataacctt
```

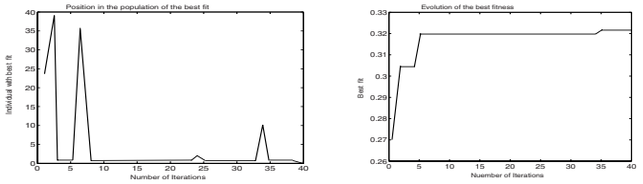
Eliminación = 0.1000, Sustitución = NO, Inserción = 0.1000
When there is initial bias and no substitution, the best alignment after 2 iterations is:

```
Op_S1,R=0:.....EI..S...I.....EE...I.E.....
S1 aSVLTQPPSvSGaPGQRVTIc_tGkSSN_igaGhNvkyQLPG_tPKLLaa
S2 qSVLTQPPSaSGtPGQRVTIScsGtSSNigsstvNwyqQLPGmaPKLLiy
S3 sggfvQPgrs_lrlcvsVifSsyamyw_vqapGk_glw_v_aievqLqss
Op_S3,R=6:.E.S.....I...ESESE.....I.E....I..EI.I.....E.E
Score = 0.6902
```

the best alignment after 36 iterations is:

```
Op_S1,R=6:.....E.....E...I.E.....EE.I...E..I.....I..I.
S1 AtGCATATTAaCTTgAAAt_caatCAaAACAT_TatcTg_aATgc_at_A
S2 AcatATATTgCaTTaAAActGgcaCATgAaATAaTgtTaCgATatacGcA
S3_aGCgcAaTACCTT_AAtgaGcggCATAACtTATTca_cCcAac__GgA
Op_S3,R=36I.....E...I...E.....E.....I.....III...
Score = 0.3240
```

And the convergence process is represented in the following figures.



(a) (b)

Fig. 3. (a) Individual rank at each iteration and (b) Objective function vs. iterations

4 Conclusions

The latest improvements that we have added to our GA for multiple sequence comparison allow driving the search towards certain regions where the user expects to find "good" sequences alignments. Those regions are defined by the restriction (totally or partially) of some operations on the sequences. The restrictions are applicable during the initialization phase or during the whole convergence process. This improvement reduces the number of iterations and the convergence to local maximum. The user has to decide, based on his previous experience, where those regions are, and select the appropriate amount of P_{ins}, P_{sus} and $P_{eli.}$But, if the user wants to explore the whole search space, setting those parameters to 1 no restriction is applied.

References

[1] Huang, X.:On Global Sequence Alignment. *Computer Applications in the Biosciences* 10: 227-235, 1994. 361

[2] Thompxon, J. D., Higgins, D. and Gibson,T.:Clustal W. *Nucleic Acids Research.* 22: 4673-4680, 1995. 361

[3] Eddy, S.:Multiple Alignment using Hidden Markov Models. *Proc. Intelligent Systems for Molecular Biology.* AAAI Press.:114-120, 1995. 362

[4] Morató, C., Seijas, J.: "Genetic algorithms for DNA/RNA sequences comparison". *European Simulation Symposium (ESS'96).* The Society for Computer Simulation. Génova. Italia, 1996. 362

[5] Rocha, R., Morató, C. and Seijas, J.: "Multiple protein sequences comparison by genetic algorithm". *Applications and Science of Computational Intelligence*, S. K. Rogers, D. B. Fogel, J. C. Bezdek, B.Bossacchi, Editors,Proceedings of SPIE 3390,:99-102, 1998. 362

[6] Seijas, J., Morató, C. and Andina, D.: "Biological Sequences Analyzed by Means of Genetic Algorithms: An efficient way for Alignment ". *WSEAS,ISPRA2002,*2002. 362

[7] Goldberg, D. E.: Genetics Algorithms in Search, Optimitation and Machine Learning. *Addison- Wesley Publishing Company, Inc*,1989. 362

[8] Gary W. G. and Thomas C. W.: An Enhanced Genetic Algorithm for Solving the High-Level Synthesis Problems of Scheduling, Allocation, and Binding. *International Journal of Computational Intelligence and Applications.* 1,(19):91-110, 2001. 363

[9] Seijas, J., Morató, C. and Sanz G. J. L.: "Genetic algorithms: Two Different Elitism Operators for Stochastic and Deterministic Applications". *Parallel Processing and Applied Mathematics*, R. Wyrzykowski, J. Dongarra, M. Paprzycki, J. Wasniewski, Editors, 4th International Conference, PPAM 2001. LNCS 2328: 617-625, 2001. 363

A real application example of a control structure selection by means of a multiobjective genetic algorithm

M. Parrilla Sánchez and J. Aranda Almansa

Departamento de Informática y Automática. Facultad de Ciencias. Universidad Nacional de Educación a Distancia (UNED). 28850 Madrid. España,
{jaranda, mparrilla}@dia.uned.es

Abstract. Control problems are clear examples of multiobjective optimization. In this kind of problems a series of objectives, some of them opposed to each other, will be optimized in order to fit some design specifications.
Moreover, evolutionary algorithms have been shown to be ideal for the resolution of these kinds of problems because they work simultaneously with a set of possible solutions, thereby favoring convergence towards a global optimum. In this document we propose a way of dealing with the different objectives considered and a genetic-evolutionary algorithm that will enable some phases of the controller design to be automated.
Finally, an application example of the methods outlined will be applied to the design of a controller to reduce the sickness index on a high-speed ship.

1 Introduction

Many problems in science and engineering require the simultaneous optimization of multiple objective functions. It will therefore be necessary to optimize a function of the form $f : S \to T$ where $S \subset \Re^n$ and $T \subset \Re^m$. However, the problem is that there is not usually an element in S producing an optimum simultaneously for each of the m objective functions. This is due to a conflict among objectives. Thus, improvement in one of them gives rise to deterioration in another. Consequently, it will be necessary to reach a compromise or trade-off solution in which all the objectives are satisfied in an acceptable degree from the point of view of the design.

Evolutionary algorithms have demonstrated good behavior in the solution of multiobjective optimization problems. Evolution in these kinds of algorithms is achieved by evaluating each solution coded in the chromosome population in order to see its fitness and compare the results of this evaluation so that the most suitable solutions have a higher probability of reproducing and transmitting their characteristics to their offspring.

When comparing fitness a new problem arises. What should be compared in the treatment of multiple objectives satisfied to a greater or lesser extent by the different chromosomes of the population?

Some ways of approaching this last problem [4] will be commented on below where a new method is proposed. Lastly, this method will be applied to a specific problem.

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 369-376, 2003.

1.1 Methods based on the notion of Pareto dominance

These methods are based on the notion of Pareto dominance. Among them, those used by Fonseca and Fleming [5] or by Srinivas and Deb [8] can be mentioned.

1.2 Methods not based on Pareto dominance

Other approaches not based on the notion of Pareto dominance have been used by other authors [4], such as the weighted sum, lexicographic ordering, or the goal programming methods [3] and [6].

2 Method of variable priorities for multiobjective evolutionary optimization

The method proposed below is inspired by lexicographic ordering and the goal programming methods, and the notion of Pareto dominance. It has been used successfully by the authors in earlier works [1] and [2].

Given $S \subset \Re^n$ and $T \subset \Re^m$, the function to optimize will be of the form $f : S \to T$. Another function $g : S \to T$ will be taken as the evaluation function of the evolutionary algorithm, where the m values returned by g are obtained by normalization of the m values returned by f.

2.1 Normalization

To carry out normalization, a vector will be established with the desired goals for the different objectives (in control problems, they will be based on the design specifications); let $o = \{o_1, ..., o_m\} \in T$ be this vector. For a chromosome x, the normalized values will be:

$$g_i(x) = \frac{o_i - f_i(x)}{o_i} \tag{1}$$

By means of normalization every objective becomes comparable to another. The original problem turns into a minimization problem.

2.2 Ranking

Population ranking is done in a similar way to the lexicographic ordering method [4], i.e., comparing the fitness vectors of the chromosomes bearing in mind the priorities of the objectives. The difference is that in the current method, priorities are not fixed, and even chromosomes in the same population will have different priorities.

Let $u = g(x_u)$ and $v = g(x_v)$ be the fitness vectors returned by the evaluation function for two different chromosomes and let $\gamma(u)$ and $\gamma(v)$ be the respective permutations of their elements, arranging them in decreasing order. $\gamma_i(u)$ and $\gamma_i(v)$ will denote their *i*th elements.

Consider also the disjointed sets *C* and *D*, whose elements will be indexes of the different objective functions, created in the following way:

$$\forall i \in \{1, \quad , m\} \begin{cases} i \in C \Leftrightarrow \gamma_i(u) < \gamma_i(v) \\ i \in D \Leftrightarrow \gamma_i(u) > \gamma_i(v) \end{cases} \tag{2}$$

Definition 1 (preferability) Vector *u* is *preferable* to vector *v* ($u \succ v$), if and only if $C \neq \phi$ and also:

$$D = \phi \quad \text{or} \quad \min(C) < \min(D) \tag{3}$$

Definition 2 (equivalence) Vector *u* is *equivalent* to vector *v* if and only if $C = D = \phi$.

The following theorems, whose demonstrations are trivial, fulfill the requirements to carry out population ranking, in function of their fitness vectors:

Theorem 1. If vector *u* is not preferable to vector *v*, and both vectors are not equivalent, then vector *v* is *preferable* to vector *u*.

Theorem 2 (transitive property) If $u \succ v$ and $v \succ w$ then $u \succ w$.

The idea is that the objective farthest from its goal will have the greatest priority in each chromosome. Population ranking will be made according to fitness vector preferability.

3 A multiobjective genetic-evolutionary algorithm for selecting and tuning controllers

Evolutionary algorithms have been used successfully to solve multiple problems, among them control problems.

The algorithm presented here is one step further in the process of automation of controller design. A genetic algorithm will try to determine the best controller structure to use.

Two loops can be distinguished inside the algorithm. One is external, consisting of a genetic algorithm with chromosomes made up of binary numbers and using the mutation and crossover operators in an attempt to determine the best control structure. The other is internal, consisting of an evolutionary algorithm whose chromosomes will be made up of real numbers and using operators adapted to this kind of codification (see [7]) that will take charge of evaluating each one of the controller structures obtained by the external algorithm.

3.1 Internal loop

Figure 1 shows a typical control system diagram. The controller block will do the plant to track the input to the system, called a reference signal. Once the structure of the controller block has been determined, the tuning process will determine the values of a series of parameters for this block.

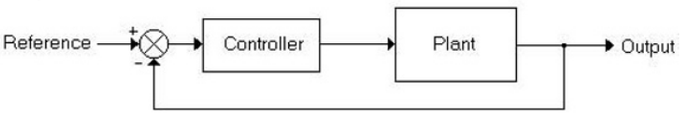


Fig. 1. Control diagram

The tuning process was carried out by the authors in [1] and [2] by means of an evolutionary algorithm, using in its fitness evaluation function the multiobjective optimization method explained in Section 2. This evolutionary algorithm will constitute the internal loop of the most general algorithm that is being presented.

Chromosomes will be coded as real number vectors, with as many elements as controller parameters need to be determined. The tournament selection technique and usual mutation and crossover operators for real number chromosomes will be used. The evolutionary algorithm will have the following steps:

1. The initial population is chosen randomly.
2. Chromosomes are decoded and evaluated by means of a computer-experiment simulation. The population is sorted according to fitness.
3. If the end conditions are given, the program concludes.
4. Tournament selection is used.
5. A new population is obtained from the selected chromosomes by means of mutation and crossover operators. In order to favor diversity and avoid premature convergence, a small number of immigrants are added (chromosomes obtained randomly).
6. Return to step 2.

This process will be repeated for each one of the controller structures determined by the external loop.

3.2 External loop

The external loop algorithm is built as a new layer or level over the internal loop just described. The objective of the external loop algorithm is to determine, by means of genetic techniques, the best control structures.

The controller will be formed by series connection with some of the following basic control subblocks:

1. Gain: K.
2. Simple pole and zero: $\frac{p}{s+p}$ y $\frac{s+p}{p}$

3. Simple lead or lag: $\frac{s+a}{s+b}$
4. Second order pole or zero: $\frac{\omega^2}{s^2+2\xi\omega s+\omega^2}$ y $\frac{s^2+2\xi\omega s+\omega^2}{\omega^2}$
5. Notch: $\frac{s^2+2\xi_1\omega s+\omega^2}{s^2+2\xi_2\omega s+\omega^2}$

Chromosomes will be made up of as many genes as different basic subblocks and each gene will be associated to one of these subblocks. Genes will take binary values, so that a 1 will indicate the presence of the corresponding subblock in the controller and a 0 will indicate its absence.

Every structure built from basic subblocks -external loop chromosomes- will be passed to the internal loop for evaluation. The internal loop will build the controller by decoding the external loop chromosome and will obtain the best values for the controller parameters. In the external loop algorithm:

1. The initial population is randomly obtained.
2. Chromosomes are evaluated by the internal loop. Population is arranged according to fitness.
3. If the end conditions are given, the program concludes.
4. Tournament selection is used.
5. A new population is obtained from the selected chromosomes by applying mutation and binary crossover operators. In order to favor diversity and avoid premature convergence, a small number of immigrants is added (chromosomes obtained randomly).
6. Return to step 2.

In short, the external loop will select the controller, while the internal loop will tune it.

4 Application example

In the CRIBAV project (Robust and Intelligent Control for High-Speed Crafts), the aim is to make robust controllers acting on some control surfaces (Flap and T-foil, see detail in Figure 2) absorb pitch and heave movements of a high-speed passenger ship. Additional information can be found on the Web: http://ctb.dia.uned.es/cribav/.

The aim will be to reduce passenger sickness index. In order to achieve this, it will be necessary to reduce the vertical accelerations of the ship due to waves. The following expression will be used for vertical acceleration, measured 40 meters away from the gravity center of the ship:

$$acv40(t_i) = a_{VH}(t_i) + a_{VP}(t_i) = \frac{d^2 heave(t_i)}{dt^2} - 40 \cdot \frac{\pi}{180} \cdot \frac{d^2 pitch(t_i)}{dt^2} \quad \textbf{(1)}$$

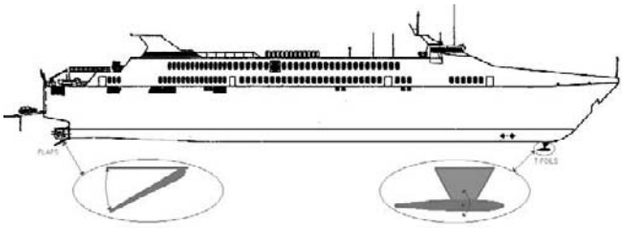


Fig. 2. Control surface detail

From the previous expression, the mean acceleration will be calculated for an experiment (by means of computer simulation). Considering N sampling instants, the expression for the mean acceleration will be:

$$J = \frac{1}{N}\sum_{i=1}^{N} \left| acv40(t_i) \right| \tag{2}$$

This expression will be employed by the evaluation function of the evolutionary algorithm to obtain a suitability measurement for a specific controller.

Different objectives considered in the evaluation function were:

Stability of the system Plant + Controller.
Saturations in the Flap: $0º \leq value \leq 15º$.
Saturations in the Tfoil: $-15º \leq value \leq 15º$.
Mean acceleration: *J*.

Because of the high computational cost required, the algorithm was implemented in C and parallelized using the specialized library MPI. The algorithm is easily parallelizable, bearing in mind that each chromosome of the external loop population can be evaluated by a different processor, and the results obtained can then be grouped. The program was executed on a Silicon Graphics Origin 2000 computer, using 30 processors, one for each external loop chromosome.

The program, once it had been designed, was executed for three different speed and sea state conditions. The sickness index reduction of the ship without control appears in the following table for different speed and sea state conditions:

Speed (Knots)	Sea State	Sickness index reduction
20	4	12,7%
40	4	46,4%
40	5	13,6%

The best results were obtained for speed = 40 knots and sea state 4. For these conditions, the controller obtained is shown in Figure 3, and the acceleration evolution and the MSI (Motion Sickness Incidence) for a range of encounter frequencies, with and without control, are shown in Figure 4.

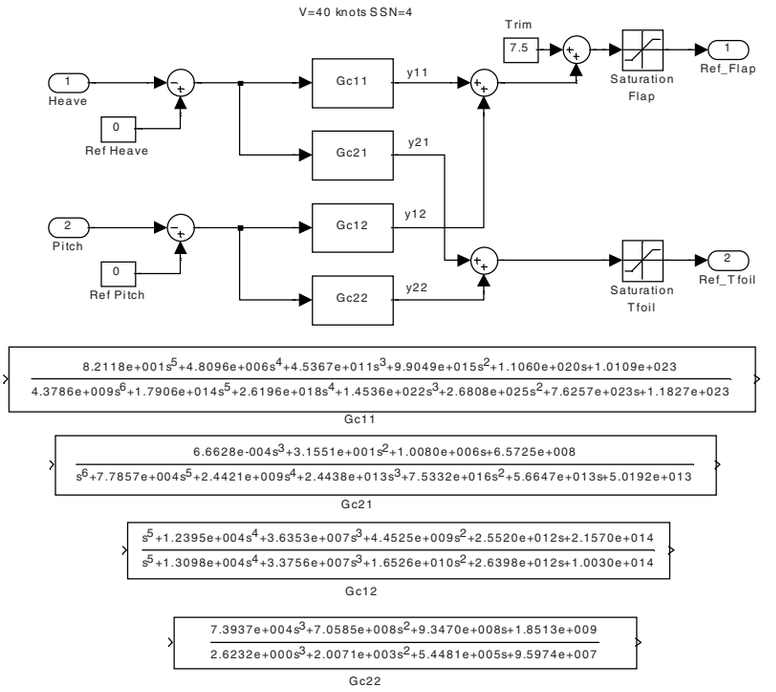


Fig. 3. Controller obtained for V = 40 knots and sea state 4.

Acknowledgment

Part of this development was supported by MCyT of Spain under contract DPI2000-0386-C03-01, and the publication is supported by a special action of the "Research Promotion Plan of UNED".

The authors also would like to thank Rafael López from the *Centro de Supercomputación de la Universidad Complutense de Madrid* for his support and the use of the university's facilities.

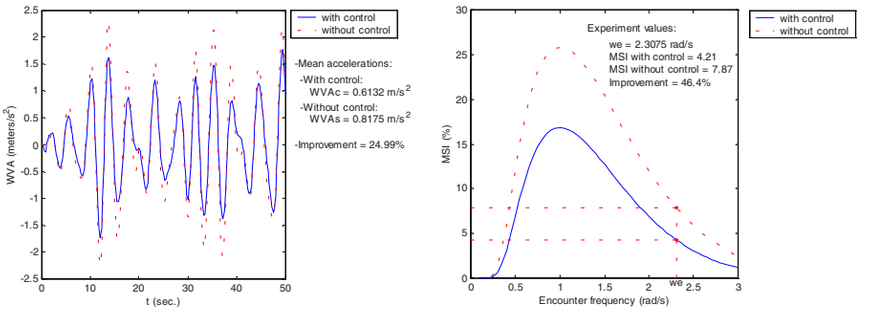


Fig. 4. Evolution of the acceleration and MSI for a range of encounter frequencies

References

1. Aranda, J.; de la Cruz, J. M.; Parrilla, M., and Ruipérez, P.: *Design of a Linear Quadratic Optimal Control for Aircraft Flight Control by Genetic Algorithm.* Controlo'2000: 4th Portuguese Conference on Automatic Control. Guimarães-Portugal. October-2000. Conference Proceedings.
2. Aranda, J.; de la Cruz, J. M.; Parrilla, M., and Ruipérez, P.: *Evolutionary Algorithms for the Design of a Multivariable Control for an Aircraft Flight Control*. AIAA Guidance, Navigation and Control Conference and Exhibit. Denver-Colorado, USA. August 2000. Conference Proceedings.
3. Charnes, A., and Cooper, W. W.: *Management Models and Industrial Applications of Linear Programming*, volume 1. John Wiley, New York, 1961.
4. Coello Coello, Carlos A.: *A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Tecniques*. Knowledge and Information Systems. An International Journal, 1(3): 269-308, August 1999.
5. Fonseca, Carlos M., and Fleming, Peter J.: *Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms-Part I: A Unified Formulation*. IEEE Transactions on Systems. Man and Cybernetics. Part A: Systems and Humans. Vol. 28. No. 1, January 1998.
6. Ijiri, Y.: *Management Goals and Accounting for Control*. North-Holland, Amsterdam, 1965.
7. Michalewicz, Z.: *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag Berlin Heidelberg. 1994.
8. Srinivas, N., and Deb, K.: *Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms*. Journal of Evolutionary Computation, Vol. 2, No. 3, pages 221-248. 1994.

Weighting and Feature Selection on Gene-Expression data by the use of Genetic Algorithms

Olga M. Pérez[1], Manuel Hidalgo-Conde[1], Francisco J. Marín[2] and Oswaldo Trelles[1,3,*]

[1] Computer Architecture Department, University of Malaga, 29080 Malaga, Spain

[2] Electronic Department, University of Malaga, 29080 Malaga, Spain

[3] Integromics, Parque Científico de Madrid, 28049 Madrid.

{olga, mhc, ots}@ac.uma.es, fjmarin@uma.es, www.integromics.com

[*] To whom correspondence should be addressed

Abstract. One of the most promising approaches for gaining insight into the biological activity of genes is to study their expression patterns in a variety of experimental conditions and contexts. In this work we present a genetic-algorithm-based approach for optimizing weighting schemes of variables used to improve clustering solutions. The same technique is used for feature selection and the detection of marker components in large datasets. An original string representation based on real numbers is used to encode the variable weight, and a modified silhouette value is used as fitness function. The strategy has a generic and parametric formulation, and effectiveness is demonstrated on gene-expression data.

1 Introduction

The spread of gene-expression monitoring technology allows the analysis of several thousands of genes simultaneously by measuring their transcription levels under different conditions, different times and from different tissues. The set of expression profiles for the genes form a table generally known as the gene-expression matrix [3].

The preferred applications for processing these data are (a) clustering analysis [6, 20, 11] based on the expression profiles of genes (or experiments) used to identify similarities in the gene behavior from which it is possible infer that those genes are co-regulated genes or, in general to understand the underlying biological processes; and (b) classification to places an object in one and only one of *a priori* defined groups. The latter is achieved by feature set selection (FSS): the reduction of large number of variables to small informative components [19].

In this work, we first focus on improving clustering solutions by using a weighting schema on the distance metrics [7]. Choosing the correct weights enhances cluster definition, both the homogeneity of the cluster itself, and separation between clusters. We then turn on attention to FSS from gene-expression data. Selecting the most important features provides both a better representation of the data and a class predictor. In this case the interest is focused on determining the best genes by their discriminative ability to locate a sample in a given group.

The strategy we propose is based on a generic application which explores the solution weighting space through a genetic algorithm. The approach is illustrated with

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 377-384, 2003.

results from the optimization of the weights for different data sets, in which we will show how the method produces noticeable improvements in cluster definition and in the selection of variables as class predictors which improve classification accuracy.

2 System and Methods

2.1 The Clustering Problem

The clustering problem can be stated in terms of a set of *n* objects defined by *p* variables, each one describing a feature of the object. Distance measurements are used for dividing the set of objects into groups or clusters, in a way that satisfies two basic criteria: homogeneity (elements in the same cluster are highly similar to each other), and separation (elements from different clusters have low similarity to each other).

Weighting a variable means giving it greater or lesser importance than others when using it to compute the distance between objects. In this work, two weighted similarity metrics will be discussed: the squared Euclidean distance and the Pearson correlation coefficient [17, 5, 15]. These metrics are general enough to allow the deduction of other similarity metrics from them.

A modified version of the silhouette value [16] is proposed to evaluate the quality of the solution [4, 13, 18]. For each object i an index $s_i \in [-1,1]$ measures the relation between b_i and a_i; where a_i is the average dissimilarity of object i to all other objects in its own cluster A and b_i is the average dissimilarity of object i to all objects in the nearest cluster B, being |A| and |C| the numbers of elements in clusters A and C.

$$a_i = \frac{1}{|A|-1}\sum_{j\in A, i\neq j} d(i,j), \qquad b_i = \min_{C\neq A}\left\{\frac{1}{C}\sum\nolimits_{j\in C} d(i,j)\right\}, \qquad S_i = \frac{b_i - a_i}{\max\{a_i, b_i\}}, \text{(Eq.1)}$$

Values of s_i close to 1 mean that object *i* is closer to its own cluster than to its neighbor cluster and is taken as well classified. On the converse relationship, values of s_i close to -1 mean that object *i* is misclassified. Kaufman and Rousseuw [12] consider a reasonable classification is characterized by silhouette above 0.5 and values below 0.2 should be interpreted as a lack of substantial cluster structure.

2.2 The Classification Problem

A classification procedure involves the assignment of objects into mutually exclusive and exhaustive groups on the basis of a set of independent variables. To this end, a linear combination of variables is proposed for discriminating between the *a priori* defined groups in such a way that the misclassification rates are minimized. We propose an automatic procedure for defining a class-predictor based on the expression levels of a sub-set of genes. In this case we work with the transpose matrix in which experiments or samples become the rows and genes the columns or variables. The problem lies in choosing the best set of genes to discriminate between classes.

Different approaches have been proposed for accomplishing FSS on gene-expression data. Golub [9] proposed a class-predictor based on a subset of genes. Briefly, they define an idealized expression pattern, corresponding to a gene that is uniformly high in one class and uniformly low in the other. A similar approach to FSS in large scale medical risk prediction problems has been used by Lowell [14], based

on expected discrimination to select features in two-class prediction problems. Alizadeh [1] used DNA micro arrays to conduct a systematic characterization of gene expression in Diffuse Large B-cell Lymphoma (DLBCL). Hierarchical clustering was used to group genes and two DLBCL subgroups, thus genes relate each subgroup to a separate stage of B-cell differentiation and activation. Bittner [2] reported the discovery of a subset of melanomas identified by analysis of gene expression in a series of samples, being able to identify subtypes of cutaneous melanoma and predict phenotypic characteristics. Iyer [11] studied the response of human fibroblast to serum, by observing the temporal program of transcription that underlies this response in 8600 human genes. A subset of genes whose expression changed substantially in response to serum was clustered hierarchically. Many features of the transcripts were related to the physiology of wound repair, suggesting that fibroblast plays a rich role in this complex multi-cellular response.

Our approach will be able to manage the problem of selecting the most relevant features for the multiple groups problem. The challenge is that the number of features (i.e. the number of genes) is especially high, producing an extremely large solution space that we propose to explore by using a genetic algorithm.

2.3 The Genetic Algorithm

Genetic Algorithms (GA) are stochastic global search methods which mimic natural biological evolution by working through a population of potential solutions applying the principle of survival of the fittest to produce closer approximations to a solution [8]. At each generation, a new set of approximations is created by selecting individuals according to their fitness, and breeding them together using operators borrowed from natural genetics. In our case, individuals represent putative solutions for the problem of optimising the property weights. We assume a collection of real property vectors (*pFile*) for which a partial solution is known (*rFile*). The goal can be stated as: search for the property weights that optimise a given fitness function. Different indices have been implemented to assess how potential solutions perform in the application domain: *silhouette* value [17], BD-index [4], Minkowski measure and Jacard coefficient [18]. A pseudo-code for the *pGA* is shown in Table 1.

Table 1. Pseudo-code outlining the parametric GA described in the main text

```
[1]  GetParameters ( pFile, rFile,oFile, I, N,E, G, M,L,C,F,D, S,T,B)
[2]  CreateInitialPopulation P_g=0(I,N,E);
[3]  for generation g=1 to G, do {
[4]          EvaluatePopulation(P_g-1,I,F,D);
[5]          Select (P_g from P_g-1, S,T)
[6]          Crossover(P_g, C);
[7]          Mutate(P_g, M,L);
[8]          if (ElitistStrategy) Elitist(P_g, B);  }
```

A population P_0 of I individuals representing property weight vectors (with N features), is initially set (step 2) with random real values in the range [0,1), satisfying: $\sum_{j=1}^{N} P_{i,j} = 1, \forall i = 1,...I$ (*Eq*.2)

In step [4] the objective function is used to evaluate each potential solution. The original description of the silhouette value [16] depends on the number of elements in each cluster, which can lead to evaluation faults. For instance, given two neighbour clusters A and B, such as the number of elements in A is greater than in B, *i.e*, |A|>>|B|, and let *i* be an object belonging to cluster A, located in the border of A and near to cluster B To manage this problem we reformulate the silhouette value as follows: Let N be a set of n elements described by *p* real-value properties. Let $C=\{C_1,C_2,.....,C_k\}$ be the partition of the set N into k exclusive clusters (k < n) and let $\overline{C}=\{\overline{C_1},\overline{C_2},....,\overline{C_K}\}$, the centroids of each cluster.

Let i∈ N, and be C_p the group to which this element belongs. The distance from element i to the centroid of the cluster C_p is $d(i,\overline{C_p})$, and the distance to the closest group for the element i is computes as the distance from i to the centroid of the cluster C_r different of C_p: $\min_{\overline{C_r}}\{d(i,\overline{C_r})\}\ \forall\ r\in\{1,..,k\}$ and $r\neq p$. Being C_q the nearest cluster, $d(i,\overline{C_q})=\min_{\overline{C_r}}\{d(i,\overline{C_r})\}$, then the quality value for element *i* is defined as: and $Q_i=\frac{d(i,\overline{C_q})-d(i,\overline{C_p})}{\max\{d(i,\overline{C_q}),d(i,\overline{C_p})\}}$ finally, the fitness value is the average of the Q_i for all objects *i* in the data set: $F=\frac{\sum_{i=1}^{n}Q_i}{n}$, that satisfy $-1\le F\le 1$.

Function *F* is used as the objective function and by maximizing this value the clustering results can be improved. The algorithm proceeds selecting individuals for reproduction (step 5) on the basis of their relative fitness. The default selection strategy works by a *roulette wheel* mechanism selecting individuals on a probability basis of the relative fitness value of individual, in such a way that the number of offspring will be proportional to the individual performance. An alternative selection procedure based on uniform sampling, has also been implemented.

In step 6, *I/2* pairs of individuals are selected to be mutually crossed to produce *I* new individuals that have part of both parent's genetic material. The number and length of the fragments to be interchanged is controlled by parameters (allowing multi-point crossover). Both fragments must be normalised to satisfy Eq.2,

Finally, mutation is applied in step 7, to produce a new genetic structure, acting as a safety net to recover good genetic material that may be lost through the action of selection and crossover [8]. Mutation is controlled with two parameters: (a) the number of individuals (mutation rate) and the number of property positions to be mutated (mutation level). Both, M*I individuals and the property positions (*j* and *k*) for the individual *i* are randomly selected. The mutation is achieved by perturbing the e_{ij} and e_{ik} positions with a ε value in the range $\max(-e_{ij}, e_{ik}-1) \bullet \varepsilon \bullet \min(1 - e_{ij}, e_{ik})$ to satisfy Eq.2.

At this point a new population has been produced by selection, crossover and mutation of individuals from the old population, and a new generation can start with the fitness evaluation for this new population. In some cases it is desirable to deterministically maintain one or more of the best-fitted individuals through the use of an *elitist strategy*, controlled by *B* parameter in step 8. The algorithm works during *G* itera-

tions. This ending has been chosen because in some cases the fitness may remain constant for a number of generations before a new superior individual is found. Finally, we mention that there is no algorithmic constraint with respect to the property vector employed [15]. In particular, it may accommodate either homogeneous or heterogeneous property vectors.

3 Results

In this section we will demonstrate the functionality of the proposed algorithm on two detailed tests, both related to gene expression data. In the first test the automatic selection of the most important features will be used to produce a class-predictor able to determine the class of new samples. The second test will demonstrate the improvement in separation and homogeneity, the two key characteristics of cluster quality.

3.1 Feature Selection and Class Predictor

For the first case the human cancer gene expression data set described in Golub [9] was used. This data set consists of 38 samples obtained from acute leukaemia patients and containing probes for 7129 human genes. Samples correspond to two different types of disease: acute myeloid leukaemia (AML) and acute lymphoblastic leukaemia (ALL). For this problem the phenotype cluster solution is known a priori: number of classes and composition. The challenge is to identify those genes that better discriminate the clusters.

Genes with low-magnitude expression values throughout all the samples do not have interesting features for discrimination purposes. Thus, the data set was normalized and a standard deviation filter was used to discard genes with flat expression levels. The final set was composed of 3699 genes. The Q-value for this data set with uniform weights and Pearson correlation coefficient was 0.376 (using Euclidean distance was up to 0.209). This value will be the baseline for this test.

The weighting schema was optimised using the GA, with parameters G=8000, L=2 (the rest were default values) and the a priori cluster information (two clusters, AML and ALL). Results are shown in Fig.1 for the evolution of the fitness value and the weight scheme. The quality of the cluster solution is improved obtaining a Q-value up to 0.844 using Pearson distance and up to 0.638 with Euclidean distance. Improvement of the silhouette value is around 74%.

Golub [9] choose the 50 genes most closely correlated with AML-ALL distinction in the known samples and use this selection as class-predictor to classify an independent data set composed of 34 samples (http://www-genome.wi.mit.edu/cgi-bin/cancer/datasets.cgi) obtaining only 6 errors. The Q-value for this data set is 0.733 (Pearson distance) and 0.556 (Euclidean distance). Equivalently, we selected the 50 genes with higher weights produced by the GA, with a Q-value up to 0.82 (Pearson distance) and 0.658 (Euclidean distance), representing a better definition for both sets. Firstly we repeat the one-left-out cross-validation procedure used by Golub et al. 1999 obtaining prediction rates up to 97% (37 of 38 hits) using between 30 and 50 genes, which compares positively with 36/38 reported by Golub [9] procedure.

Using these 50 genes as class-predictor for the independent set, we were able to predict 28/34 samples based solely on the 39 genes with highest weights. The separa-

tion between centroids of each class (ALL and AML classes) using unweighted and weighted Euclidean distances were 0.075 and 0.112, respectively, thus a noticeable and better separation between classes is obtained when using a weighted schemes.

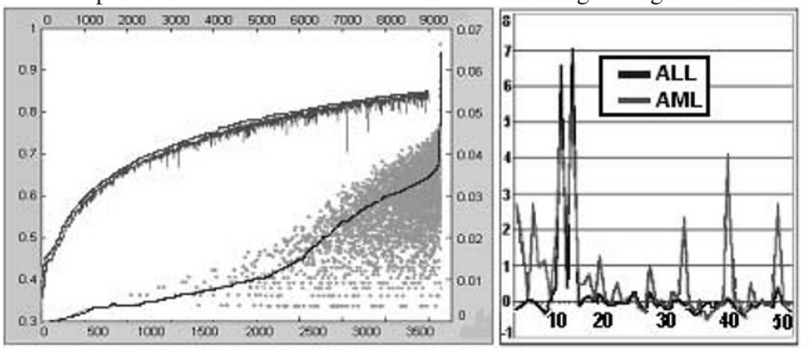


Fig. 1: Silhouette evolution (on the left) for max and min fitness value. Improvements from initial values of 0.4 to up to 0.8 (vertical left-hand scale), from 8000 iterations (upper horizontal scale). Weighing schema for the 3699 genes (bottom horizontal scale) obtained for 10 runs (vertical scale on the right hand side). We have used logarithmic scales for a clearer presentation of the weighting scheme. On the right the average gene expression levels in classes AML and ALL for the 50 best genes ordered by weight

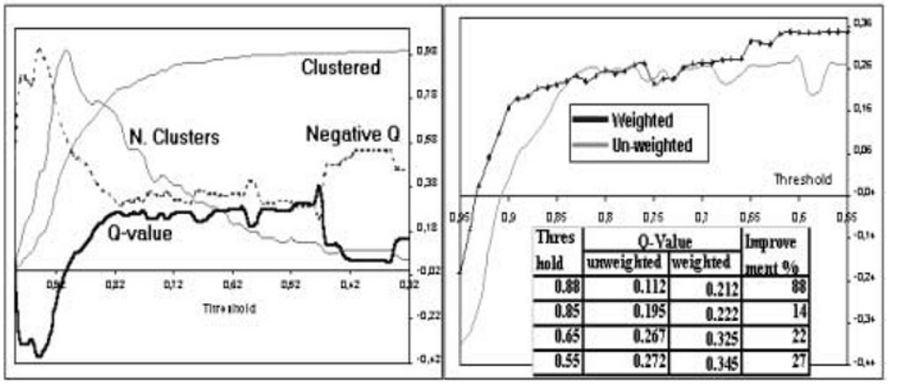
3.2 Improving Clustering Solutions

The second case we present is based on the study of the response of human fibroblast to serum. Iyer [11] used DNA micro array hybridisation to measure the temporal changes in mRNA levels of 8613 human genes at 12 times after serum stimulation, identifying a subset of 517 genes whose expression levels substantially changed in response to serum (www.sciencemag.org/feature/data/984559.shl). Genes were hierarchically clustered by the procedure of Eisen [6], and discrete clusters were obtained by hand inspection using biological knowledge of human experts. The quality of such manual-solution is evaluated by our proposed *Q*-value was 0.26.

We will proceed in the following way: the original data set will be hierarchically clustered using uniform weighting scheme, and discrete clusters will be produced by using a cut-off threshold. Since the election of the appropriated threshold is out of the scope of this document, we will proceed by moving the threshold in the whole distance range. For each solution the quality of the clustering will be evaluated among the following parameters: number of clusters and elements being clustered (those with distances between elements above the threshold), the proposed Q-value and the number of elements with negative silhouette value (see Fig.2, on the left). Next, a weighting scheme will be obtained by using the proposed GA, and the same experiment will be repeated, now using the obtained weights. We will show that cluster solutions are improved when using the weighting scheme.

The initial partial solution, used as seed-set to evolve the weight scheme through the GA) is composed by the strongest related objects (i.e. more than 0.94 of correlation index). Using this value as threshold 28 discrete clusters were produced from the

hierarchical tree, reporting an initial *Q*-value of 0.8. The weighting scheme was optimized using the GA over the partial cluster solution. In this case each individual in the population represents alternative solutions for the weights assigned to each variable. The algorithm applies the solution and computes the new clustering distances using that solution. The parameters used were: B=2, G=300 and Pearson correlation distance, the rest were default values. The initial data set was hierarchically clustered using the weighting scheme defined by the GA from the seed set. In Fig. 2 (on the right) we shown the evolution of the Q-value as a function of the threshold compared to the analogous clustering results obtained using homogeneous weights. An important improvement in the quality of the clustering solution can be observed, in practical terms in the whole range of thresholds.



Threshold	Q-Value unweighted	Q-Value weighted	Improvement %
0.88	0.112	0.212	88
0.85	0.195	0.222	14
0.65	0.267	0.325	22
0.55	0.272	0.345	27

Fig. 2 On the left, the evolution of the proposed parameter to evaluate the cluster solution as a function of the threshold value used to transform a hierarchical solution into discrete clusters. The *Q*-value as defined in the main text; the number of objects with negative Q-value (misclassified); the number of clusters formed and the total number of classified objects are presented. On the right, the Q-value as a function of cut-off thresholds and the improvement percentage in cluster quality. In practically all cases weighted clustering schemes produce better Q-values, which means better cluster solutions.

4 Discussion

Genetic Algorithms are stochastic global search methods that mimic natural biological evolution to devise solutions for large optimization problems. This is the case for class-prediction methodologies which require selection of a representative set of markers from a large number of variables. Weighting schemes have also proved to be a useful technique for enhancing cluster definition, both the homogeneity of the cluster itself, and the relation between clusters.

In this work we have presented a generic application for optimizing the weighting scheme used to compute property distances that can also be applied in the FSS domain. The application has been tested in two exhaustive exercises, achieving in both cases a substantial improvement in the quality of results. Our algorithm should be considered as a sort of automatic tool for determining, at an affordable computational cost, the best combinations of weights to be used for computing clustering distances.

Acknowledgement

This work has been partially supported by grant QLRT-2000-01473 of the EU programme and it was developed during the stage of Ms. Olga Perez at the EMBL-EBI under the Marie Curie program (QLRI-1999-50595). Integromics is a partner of the University of Málaga in Information Technology applied to Life Sciences.

References

1. Alizadeh, A.A.; *et. al*. (2000) "Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling" Nature 403, 503-511.
2. Bittner,M.; *et. al*.(2000) "Molecular classification of cutaneous malignant melanoma by gene expression profiling" . Nature 406, 536,540.
3. Brazma, A. and Vilo J. (2000), "Gene expression data analysis", FEBS letters, vol 480, Issue 1, pp 17-24
4. Davies, D.L. and Bouldin, D.W. (1979), "A cluster separation measure", IEEE Trans. Patt.Anal. Mach. Intell. 1 pp.224.227
5. Dillon, W.R. and Goldstein ,M. (1984) "Multivariate Analysis: Methods and Applications". John Wiley & Sons, New York .
6. Eisen, M., Spellman, P.T., Botstein, D. and Brown, P.O. (1998) Proc. Natl. Acad. Sci. USA 95, 14863-14867
7. Everitt, B. (1993), "Cluster analysis", London: Edward Arnold, third edition.
8. Golberg, D.E., (1989), "Genetic Algorithms in Search, Optimisation and Machine Learning", Addison Wesley Publishing Company.
9. Golub,T.R. et.al. (1999) "Molecular Classifications of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring". Science 286:531-537
10. Hartigan, J.A., (1975), "Clustering Algorithms", Wiley, New-York
11. Iyer,V.R.; et.al (1999) "The transcriptional program in the response of human fibroblast to serum" . Science 283 (5398):83-87.
12. Kaufman, L. and Rousseeuw, P.J. (1990). "Finding groups in data. An introduction to cluster analysis". Wiley-Interscince, New York.
13. Jain, A,K, and Dubes, R.L. (1998), "Algorithms for clustering data", Prentice-Hall
14. Lowell, D.R.; et al. (1997) "On the use of expected attainable discrimination for feature selection in large scale medical risk prediction problems". CUED/F-INFENG/TR299
15. Perez, O. M.; Marin F. J.; and Trelles, O. (2001), "Improving Biological Sequence Property Distances by using a Genetic Algorithm", IWANN 2001, LNCS 2085, pp. 539-546.
16. Rousseeuw, P.J. (1987) "Silhouettes: A graphical aid to the interpretations and validation of cluster analysis". J. of Computational and Applied mathematics,20:53-65.
17. Ríos Sixto (1983), "Análisis estadístico aplicado". Madrid: Paraninfo,1983. 3ª edición.
18. Sokal , R.R. (1977), "Clustering and classification: background and current directions", In Van Ryzin, J. ed., Classification and Clustering, 1-15, Acad. Press.
19. Stefanini, F.M. and Camussi, A. (2000) "The reduction of large molecular profiles to informative components using a genetic algorithm" Bioinformatics 16, 923-931
20. Tamayo,P.; et.al. (1999) "Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation". Proc. Natl. Acad. Sci. USA 96 (6),2907-2912.

Supervised Segmentation of the Cervical Cell Images by using the Genetic Algorithms

Nadia Lassouaoui[1], Latifa Hamami[2] , and Fairouz Chehbour[1]

[1] Centre de Recherche sur l'Information Scientifique et Technique
Computer System Laboratory
3 rue des frères Aissou , Ben Aknoun, Algiers, Algeria.
{nlassouaoui, fchahbour}@mail.cerist.dz
[2] Ecole Nationale Polytechnique
Electronic Department. Signal & Communications Laboratory.
B.P. 182 16200 El-Harrach, Algiers.
l_hamami@hotmail.com

Abstract. This paper deals with the separation problem of cervical cell image on its core image and its cytoplasm image. This segmentation permits the morphological analysis of each component for deducing a decision about the malignity of the cell. For that, we develop a supervised algorithm based on genetic algorithms. Firstly, we use an iterative algorithm for finding the thresholds which delimit the various classes of the image (core, cytoplasm and background). Then, all points of each class evolve in an evolutionary process based on genetic algorithms, this step permits to find the true class of each point. So, we obtain more precise results. We note that the core images and the cytoplasm images constitute the data base of a recognition stage for the vision system for tracking the cervical cancer. We applied our algorithm on several images, herein, we present some results obtained by two cervical cell images.

1 Introduction

To recognize the malignant cells, it appeared judicious to divide them into their various components, for analyze the morphology of each one for deducing a decision about the malignity of each cell [5]. The cell images are divided mainly into three components : the core which is the heart of the cell, the cytoplasm which represents all the part of the cell that surrounds the core, and the background which surrounds the cells. For doing this separation, we must use an algorithm of segmentation in homogeneous areas. In the literature ([2], [6], [8]), the segmentation consists in detecting the objects or classes of objects that constitute an image. This decomposition is one of the most difficult tasks of the machine analysis of images ([2], [6], [8]), in particular when they are medical images. There are various methods ([2], [7], [10], [11]), but there is no method that is a solution of all types of images. Herein, we develop and present a supervised algorithm based on genetic algorithms GAs. In the following section, we present the basic concepts of GAs, in section 3, we

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 385-392, 2003.

present our algorithm, in section 4, some application results. Finally, we give the conclusion.

2 Basic Concepts of GAs

The basic principles of GA were first proposed by John Holland ([3], [4], [9], [12]). GA is inspired by the mechanism of natural selection, a biological process in which stronger individuals are likely be the winners in a competing environment.
Throughout a genetic evolution, a fitter chromosome has the tendency to yield good-quality offspring, which means a better solution to the problem. In a practical application of GA, a population pool of chromosomes has to be installed and they can be randomly set initially. In each cycle of genetic operation, termed an evolving process, a subsequent generation is created from the chromosomes in the current population. This can only be successful if a group of those chromosomes, generally called "parents" or a collection term "mating pool", are selected via a specific selection routine [4]. The genes of the parents are to be mixed and recombined for the production of offspring in the next generation. It is expected that from this process of evolution (manipulation of genes), the "better" chromosome will create a larger number of offspring, and thus has a higher chance of surviving in the subsequent generation, emulating the survival-of-the-fittest mechanism in nature.
The cycle of evolution is repeated until a desired termination criterion is reached. This criterion can also be set by the number of evolution cycles, the amount of variation of individuals between different generations, or a predefined value of fitness [3].
In order to facilitate the GA evolution cycle, two fundamental operators "crossover-mutation" are required. Crossover exchanges portions of the population chromosomes (recombination process) to obtain another new ones (see figure 1).

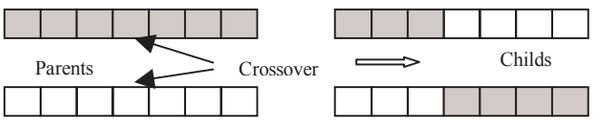


Fig. 1. Example of one crossover point.

However, for mutation, the process is applied to each offspring individually after the crossover exercise. It alters each bit randomly with a small probability.
The following pseudo-code summarizes the standard genetic algorithm.

```
Standard Genetic Algorithm
{       t:=0; //start with an initial time
        init_populaion P(t) //initialize an usually random population of individuals
        evaluate P(t) //evaluate fitness of all individuals of population
        While not done do //test for termination criterion (time, fitness, etc)
        begin
        t:=t+1;
```

```
P':=select_parents P(t); //select a sub-population for offspring production
Recombine P'(t); //recombine the genes of selected parents
Mutate P'(t); //perturb the mated population stochastically
Evaluate P'(t); // evaluate its new fitness
P:=survive( P, P'(t)); //select the survivors from actual fitness
End;}
```

In our work, we exploit this standard GA for developing a supervised algorithm for segmentation in homogeneous areas of images.

3 Supervised Segmentation based on GAs

This algorithm contains three main steps that are:

- Find the thresholds that delimit all classes of the image.
- Learning using the GAs.
- Classification stage for finding the final segmented images.

Now, we give more details about each step.

3.1 Supervised search of the various classes of the image

For finding the various classes of the image, we use an iterative and a recursive algorithm ([2], [6], [8]), it is based on the computation of the histogram of the image. For separate the object from the background in an image, we must find a threshold S that is the grey level limit between the objects and the background, for that we follow this algorithm:

1. Compute the histogram H of the image. $H[i]$ gives the frequency of the grey level i, i.e., the number of the points having a grey level equal to i.
2. Compute the mean T of the image, it is considered as the initial threshold.
3. Compute the report $T_{object}=t_1/\ 2*t_2$, with:

$$t_1=\sum_{i=0}^{T} i*H[i] \qquad \text{and} \qquad t_2=\sum_{i=0}^{T} H[i] \tag{1}$$

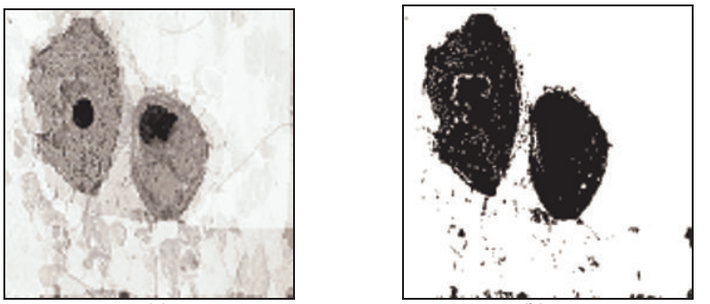
4. Compute the report $T_{fond}=t'_1/\ 2*t'_2$, with:

$$t'_1=\sum_{i=T+1}^{Max} i*H[i] \qquad \text{and} \qquad t'_2=\sum_{i=T+1}^{Max} H[i] \tag{2}$$

   with Max: is the maximal intensity of an image.
5. If the threshold T is equal to $(T_{object}+T_{fond})$, then T is the desired threshold and then $S:=T$. If T doesn't equal to $(T_{object}+T_{fond})$, then we affect to T the value $(T_{object}+T_{fond})$, and repeat the steps 3 to 5 until the $(T_{object}+T_{fond})$ will be equal to T, and then $S:=T$.

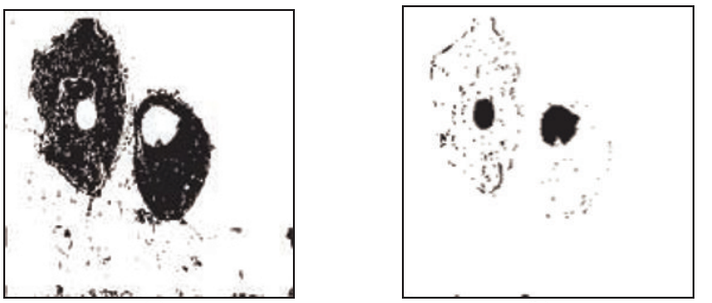
An application example is given by figure 2, for the extraction of the cells from the background. Knowing that the background is more luminous then the cells. So, for finding the second threshold S' that separates the components of the cells, we applied

the algorithm on the set of the points with intensity between *0* to *S-1*. The figure 3 presents the separation of the preceding cell image. Another example is given by figure 4. The images used are in bmp format of size 256x256.



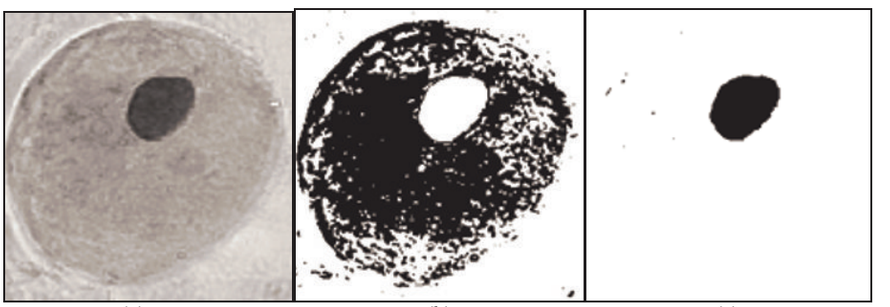
(a) (b)

Fig. 2. Extraction of the cells from the background *S*=171.



(a) (b)

Fig. 3. Separate the components of the cells *S'*=82, (a) cytoplasm image and (b) core image .



(a) (b) (c)

Fig. 4. Separate the components of the cell of figure (a), *S'*=90 and *S*=157.

We can check that appear some random points in each extracted areas, so, there are confusion between the various classes. For decreasing this confusion, we propose to add another stage based on GA that permits to seek the true class of each point.

3.2 Learning by GAs

This step must find the best chromosome that characterizes each class. For that, we adapt the standard GAs, and we applied it on each class. Herein, we give more details about each step of GAs.

Generation of initial population. Select randomly a number N_p of pixels in the class, we note that each pixel is represented by its chromosome. The chromosome (figure 5) at the point s is a vector of size 9 or 25, it contains the intensity values of pixel s and its neighbourhood a_i.

a_1	a_2	a_3	a_4	s	a_5	a_6	a_7	a_8

Fig. 5. The chromosome at the point s of size 3x3

Compute the fitness. It is a value which evaluates each chromosome. In our method, the fitness function of a chromosome is the measure of difference between the chromosome and the mean chromosome, i.e., The genes (components of chromosome) must be close to the mean. The mean chromosome is equal to the mean of all chromosomes which constitute the class. The fitness f of a chromosome of size l_chrom is merely calculated as follows :

```
function fitness
Begin
f:=0;
        For i=1 to l_chrom do
        Begin
        If |chromosome[i]–mean[i]|≤Sf  then  fit[i]=1 else fit[i]=0;
                f:=f+fit[i];
        End
End
```

S_f is a threshold, its value is given according of the image.
The chromosomes having the highest fitness values are selected, recombined and muted.

Selection. The selection is doing by the Roulette Wheel ([3], [4]). Then, for selecting a chromosome in a population of size N_p, we follow this algorithm:

```
Function selection
Begin
For i:=1 to Np do
Begin
```

Generate randomly a real number R_i, with $R_i \in [0,1]$;
If $R_i < P_1$ then
Select the chromosome of index *1*;
If not
Select the chromosome of index k: $2 \leq k \leq N_p$ such as: $P_{k-1} < R_i \leq P_k$.
End;
End;

We note that P_i is equal to $\frac{f_i}{\left(\sum_{j=1}^{N_p} f_j\right)}$, with f_i is the fitness of the chromosome of index i in the population.

The crossover. One point crossover with a probability *Pc* is used. A number k is generated randomly between *0* and *l_chrom* (*l_chrom* is the length of the chromosome). The portions of parents between this crossover point are exchanged to form new strings.

The mutation. The mutation of k^{ith} gene of chromosome with a probability *Pm* consists to replace its value by a randomly value in *[Vmin, Vmax],* where *Vmin* and *Vmax* are respectively minimal intensity and maximal intensity of the class. Such as *[Vmin, Vmax]* is equal to : *[0,S']* for the core areas, *[S',S*] for the cytoplasm areas and *[S, Max]* for the background areas.

The stopping criteria. We choose the fixed number of generations as stopping criteria for execution of GA. That means the cycle: evaluation by the compute of the fitness, selection, crossover and mutation, is repeated N times, where N is the maximal number of generations.
Finally, for each class, we obtain the best chromosome. In our application, there are three best chromosomes, each one corresponds to: core, cytoplasm or background areas. Then, we do the classification of the points.

3.3 Classification procedure

It is the final step, herein, we affect each point to its true class. To classify the point *s* requires the compute the distance between its chromosome and each best chromosomes. The pixel will be affected to the class where it has the minimal distance. For compute this distance, we adopt the same principle as the compute of the fitness, and then, the point will be affected to the class *Cl* that it has the maximal fitness *f[Cl]*. Moreover, we add another criteria that is the dispersion of each chromosome estimated by the variance.

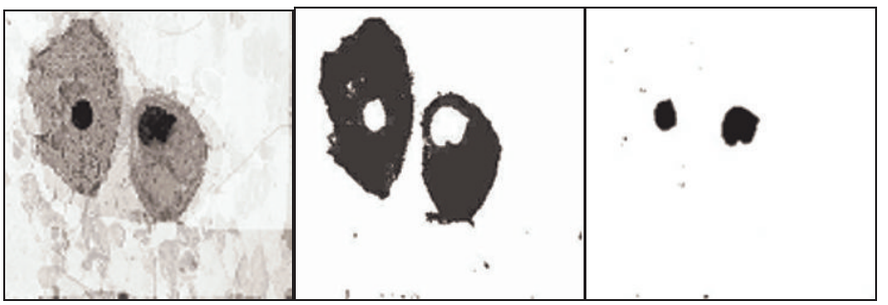
$$If \left| \mathrm{var}(chromosome) - \mathrm{var}[Best\_Chromosome[Cl]] \leq S_c \right| \ then\ f := f+1\ ; \tag{3}$$

S_c is a threshold given according of the image.

We note that our proposed algorithm doesn't call to the functions coding and decoding ([3], [4]), that are characteristics of all GAs. Then, we note a profit in time, since the chromosomes are presented by their real values.

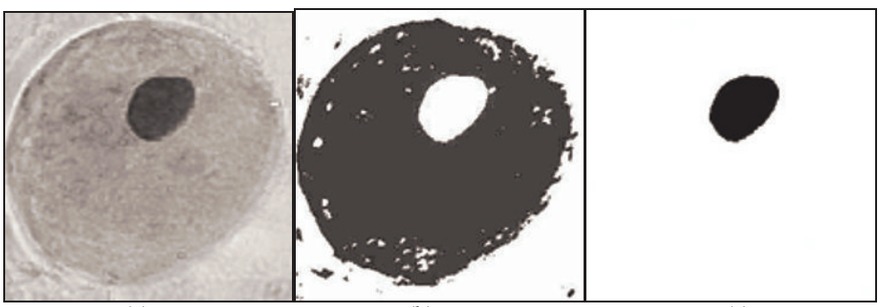
4 Application

We applied this algorithm on several images, we present by the figures 6 and 7 the results obtained by the two preceding images. We take these parameters for the GAs: *Pc*=0.6, *Pm*=0.03, *Np*=50, *N*=200.
We obtain better results that those obtained before applying the AGs (figures 3 and 4). So, there is less confusion between the areas. We note also that the extracted of the cytoplasm is often difficult, since it has no homogeneous areas and it has a texture close to that of the background. Also, the background is not a homogeneous area.



(a) (b) (c)

Fig. 6. Extracted cytoplasm image (b) and core image (c) of figure (a).



(a) (b) (c)

Fig. 7. Extracted cytoplasm image (b) and core image (c) of figure (a).

5 Conclusion

In this paper, we propose a supervised algorithm of segmentation in homogeneous areas based on genetic algorithms, we apply our algorithm on cervical cell images. Our goal is the separation of each cell on its core and its cytoplasm, so, in our segmentation, each part must correspond to the precise part in the original image. This separation allows to analyze the morphology of each component for deducing a response about the malignity of each cell. We obtain good results in a short computing time since our algorithm works directly with the intensity of the pixels, and then, it doesn't call to the coding and decoding functions, that are characteristics of all algorithm genetic.

References

1. Bhanu, B., Lee, S., Ming, J.: Adaptive Image segmentation Using a Genetic Algorithm, IEEE Transactions on Systems, Man, And Cybernetics, Vol. 25, N°12, (1995) 1543-1567.
2. Cocquerez, J.P., Philippe, S.: Image Analysis : filtering and segmentation, Masson Edition (1995).
3. De Jong, K.: Learning with Genetic Algorithms: An Overview, Machine Learning Vol. 3, (1988) 121-138.
4. Goldberg, D. E.: Algorithmes G□n□tiques, Edition Wesley (1989).
5. Gompel, C.: Atlas of cytology, Medical School of Paris (1982).
6. Gonzalez, R.C.: Digital Image Processing, second edition, Addison Wesley (1987).
7. Hamami, L., Lassouaoui, N.: A Fractal Approach and Analysis of Variogram for Edge Detection of biomedical images, Bio-Inspired Applications of Connectionism. Lecture Notes in Computer Science LNCS 2085, Part. 2. Springer-Verlag, Berlin Heidelberg (2001) 336-344.
8. Haralick, R.M., Shapiro, L.G.: Image segmentation Technique, Computer Vision, Graphics and Image Processing, Vol. 29, (1985) 100-132.
9. Holland, J.H.: Genetic Algorithms, for the science, Scientific American Edition, N°179, (1992) 44-50.
10. Lassouaoui, N., Hamami, L.: Brain Image Segmentation using mathematical morphology, 2nd IASTED International Conference on Visualization Imaging, and Image Processing, VIIP2002, Malaga, Spain, (2002) 9-12.
11. Lassouaoui, N., Hamami, L.: Segmentation of biological cell images by the use of Multifractal Approach, Ninth International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems IPMU, France, (2002) 2005-2012.
12. Man, K.F., Tang, K.S., Kwong, S.: Genetic Algorithms: Concepts and Applications, IEEE Transactions on Industrial Electronics, Vol. 43, N°5 (1996) 519-534.

Using Genetic Algorithms for solving partitioning problem in codesign.

Mouloud KOUDIL*, Karima BENATCHBA*, Daniel Dours **

*Institut National de formation en Informatique, BP 68M, 16270, Oued Smar, Algérie.
** Institut de Recherche en Informatique de Toulouse, IRIT – UPS, UMR CNRS 5505 Université Toulouse III, 118, route de Narbonne, 31062 Toulouse Cedex, France.
Email : m_koudil@ini.dz

Abstract. Partitioning problem in codesign is of critical importance since it has big impact on cost/performance characteristics of the final product. It is an NP-Complete problem that deals with the different constraints relative to the system and the underlying target architecture. The reported partitioning approaches have several drawbacks (they are often dedicated to a particular application or target architecture, they operate at a unique granularity level, most of them are manual and impossible to apply for complex systems, the number of constraints they deal with is generally limited...). This paper introduces an automatic approach using genetic algorithms to solve partitioning in codesign. This approach is totally independent of target architecture. Another advantage of this approach is that it allows determining dynamically the granularity of the objects to partition, making it possible to browse more efficiently solution space.

1 Introduction

Embedded systems are used in more and more domains of daily life. One of the most significant examples is the use of such systems in medicine: exploration tools, various prosthesis (pacemakers, hearing prosthesis...), etc. These embedded systems are made of a software part ran on a hardware architecture made of various processors: hardware processors, software set-instruction processors, etc. The concurrent design of the different parts of the system (software and hardware) is called codesign.

The problem encountered during the codesign process is the task of determining the parts of the system that must be implemented in hardware and those parts that are to be in software [1]. This task is named "partitioning" and is of critical importance since it has a big impact on final product cost/performance characteristics [2]. Any partitioning decision must, therefore, take into account system properties. It must also include number of constraints related to the environment, implementation platform and/or system functionality requirements. Partitioning is known to be an NP-Complete optimization problem. There exist different approaches to solve partitioning problem. They can be either manual or automatic.

Numerous approaches try to solve partitioning problem in codesign, but number of are manual [3, 4, 5]. Among the approaches that try to automatically solve partitioning problem, the simpler technique is the exact one [6, 7]. Exact algorithms are based on the determination and the evaluation of all possible solutions. In theory, they allow,

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 393-400, 2003.

for sure, to achieve all optimal solutions. In practice, these approaches are ideal for very small size problems, but they become intractable as soon as problem size gets larger. In fact, their main drawback is the execution time that grows exponentially with the number of system tasks to partition. Computing time becomes prohibitive as the size goes over 20. For example, the exhaustive browse of the solution space for 64 tasks mapped on two processors, with a 2 GHz computer, assuming that the evaluation time of a solution takes only one clock cycle, would take 292 years!!!

One alternative is the use of approached methods that allow getting one (or many) solutions in an "acceptable" time. There are mainly two kinds of heuristics: the methods dedicated, and the general heuristics, that are not specific to a particular problem.

Among dedicated strategies, there are [2, 8, 9]. The main advantage of applying specific approaches is that they are "tailored" for the given problem. However, the solutions become hard to deal with as soon as a change appears, even small, in the type of systems to design.

The general heuristics [10, 11, 12] are not dedicated to a particular type of problems, and are widely used in other research fields, consisting of NP-Complete problems. This class encloses: local search and evolutionary algorithms. They allow to find solutions in a short time. Their drawback is that it is impossible to guarantee that then generated solution is the optimum. In fact, that this kind of algorithms is often trapped in a local optimum, and never reaches a global optimum.

The weaknesses of actual partitioning approaches are that: some of them are dedicated to a given application and, hence, hard to generalize; they operate at a unique granularity level, either too low or too high, often missing interesting solutions; most of them are manual and difficult to apply as soon as the system size increases; they take into account a small subset of the possible constraints that apply to systems (execution time, software and hardware space, communication...); they are dedicated to a given target architecture type, and impossible to extend to other platforms.

This paper introduces an automatic approach using genetic algorithms to solve partitioning in codesign. This approach is totally independent of target architecture. Another advantage of this approach is that it allows determining dynamically the granularity of the objects to partition, making it possible to browse more efficiently the solution space.

Section two introduces the proposed partitioning approach while the third one gives more details about the automatic Multi-level partitioning approach. Section four lists some simulations and experimental results.

2 The proposed partitioning approach

When the size and complexity of the problem rise, it becomes difficult, for a human being, to apprehend all the details, and manual resolution of the problem becomes intractable. This is the reason why an environment called PARME [13] was designed, allowing the user to test partitioning heuristics. It offers the opportunity to study the parameter values, as well as the strategies to use, according to the type of problem. This work is inspired by those of Benatchba and al. [13] that propose to use general heuristics (simulated annealing, taboo search, genetic algorithms, scatter search, etc.)

for satisfiability (Sat and Max-sat) problem solving. The approach proposed here, transposes the execution schemes introduced in [13], to the resolution of partitioning problem. In this paper, we introduce the tests performed with genetic algorithms, using PARME. Such tests allow tuning the parameters and strategies of the algorithm. An exact approach is also offered to the designer for an exhaustive search of solutions. This method presents acceptable times for applications that do not exceed 20 entities mapped on 2 processors.

2.1 Representation of a partitioning solution

The partitioning problem, in our approach, can be summarized as the mapping of a certain number of entities onto the different processors of the target architecture, while trying to optimize a cost function defined by the user. The proposed method is the following:

Let ***Nbe*** be the number of entities of the application under design and ***Nbp*** the number of processors of the target architecture. The representation technique consists of creating a vector of ***Nbe*** cells, in which each entry corresponds to an entity number. The entities are sorted in the increasing order. Each vector cell contains the number of the processor to which the corresponding entity is affected during partitioning.

Example:

The following representation corresponds to a solution of a partitioning problem with 8 entities (*Nbe=8*) mapped on 4 processors (*Nbp=4*).

Entity	0	1	2	3	4	5	6	7
Processor	3	2	0	1	2	3	0	1

This means that the entities 2 and 6 are mapped on processor Number 0; 3 and 7 are mapped on processor Number 1; 1 et 4 are mapped on processor Number 2; 0 et 5 are mapped on processor Number 3;

2.2 Cost function

The partitioning algorithm is guided by a cost function that allows evaluating the quality of a given solution. It takes into account different cost constraints (hardware and software space), performance constraints (particular object execution times, global application time...) and communication (because information exchange between different application entities is often a bottleneck). The characteristics taken into account in our approach are thus: space, execution time and communication. However, a frozen cost function leads to solutions that might be interesting for some particular criteria, but prohibitive for other ones. This is the reason why different weights can be associated to the criteria of the cost function, according to the type of the considered entities. The generic cost function is given by the following formula:

$$\mathbf{F = A*}\mathit{Space} + \mathbf{B*}\mathit{Time} + \mathbf{C*}\mathit{Communication} \quad (1)$$

Where: ***Space***: is the space taken by the entity in a memory or a register; ***Time***: is the execution time of the entity on the chosen processor; ***Communication***: is the cost of the communication; and ***A, B, C*** are the weights associated to the different criteria.

An example of cost function using a combination of different parameters is illustrated in the following: **f(X,Y,Z)=X+Y*2+Exp(Z)**

In this function, the three parameters are used (X: Space, Y: Time, and Z: communication). The communication parameter is the most important (it participates exponentially to the cost function), while space is the least important.

3 The automatic Multi-level partitioning approach

In order to evaluate the different algorithms implemented in PARME, several benchmarks have been used, but for space reasons, we focus on a particular one, which corresponds to a real program: it represents the entities of an optimization algorithm used in PARME environment. The objective is to reduce the execution time of this algorithm by splitting it onto several processors, since it is, despite its rather small size, very time-consuming. This benchmark is initially composed of 7 entities (objects of the algorithm). The target architecture is composed of an instruction-set processor and a hardware processor. The cost function is a linear combination of the three main criteria (respectively: space, execution time and communication): **f(x,y,z) = x+y+z**.

The benchmark described here presents a reduced number of entities and processors, and thus of possible solutions (128). It is possible to apply an exact method to it. The result is that it exists no "feasible" solution (by "feasible" solution, we mean a solution that leads to a coherent implementation). This fact tends to prove the complexity of the benchmark, despite the rather small number of solutions it presents. The main problem that makes a solution intractable is that the sum of the spaces of the entities affected to a given processor exceeds the initial space available on it. This would be equivalent to try to affect to a knapsack, a load that exceeds its capacity.

This benchmark emphasizes the drawback of the majority of the reported approaches to solve partitioning problems: the fixed granularity (object level in this example) does not allow taking into account the intrinsic complexity of the application, often leading to no solution or bad solutions. This is why a multi-level partitioning technique is proposed in this section. This technique performs a backtracking on a refinement procedure every time no acceptable solution is found, as long as the stopping criterion is not reached. The term "***entity***" is used as a generic term specifying an object, a method or an elementary instruction bloc. The multi-level partitioning process follows the steps illustrated in fig. 1. The first one consists of building an application entity tree, only once, during the specification analysis phase. The timing, space and communication characteristics are then computed for each node of the tree.

The iterative refinement procedure replaces every entity that is declared "complex", by the nodes of the level that lies directly below, in the sub-tree whose root is this entity. An entity is declared "complex" when its cost function has a value exceeding a certain threshold that is initially fixed by the user, and that decreases during the partitioning process.

```
Step 1 : Build the entity tree;
Step 2 : Analyze the characteristics of the entities;
         Build parameter matrix (time, space, communication);
Step 3 :
    While stopping criterion is not reached do
    Perform automatic partitioning Procedure (search for a solution);
        If no solution has been found
        Then
            Perform identification procedure (identify complex entities);
            Perform refinement procedure for the entities declared "complex";
            Update entity list (entities considered during the next step
                    of the partitioning algorithm);
        EndIf
    EnDo
```

Fig. 1. General scheme of the iterative partitioning algorithm.

Tree building procedure:
In the specification, the C++ code of the application objects is composed of methods, built up of elementary blocks of code (loops, conditionals, alternatives…). We do not go down to the elementary instruction level that constitutes a grain that is too fine, and thus too complex to deal with in applications that can contain millions of lines of code. The building procedure of the entity tree proceeds as follows:

- application objects constitute the level 1;
- the different methods of all the level 1 entities form the level 2 nodes;
- the basic blocks encapsulated in the methods constitute level 3 nodes.
- All the nodes are considered as entities, independently of what the tree level is, and the same process is applied.

Identification procedure:
To be able to decide if a node is "complex", a threshold is first introduced by the user, concerning the costs involved by the different constraints (space, time, communication) and the global cost of an entity.

Any entity whose cost function has a value that exceeds the threshold is declared "complex" and is refined. However, since the costs decrease during the splitting algorithm, at a certain point of the process, all the costs become lower than the initial threshold and none of the entities can be considered as "complex". At that point, if no acceptable solution (less than a minimum cost also fixed by the user) has been found, the threshold is replaced by the average of the lowest and the highest costs of all the entities of the partitioning list.

Refining procedure:
This step consists of replacing each entity that is not a leaf, and that presents characteristics that exceed the fixed threshold, by the entities that are on the nodes of the level which is directly below in the sub-tree having the current entity as root. For example, an object would be replaced by its methods. The new nodes become individual entities stored in the list used by the partitioning algorithm. Each node of the tree that is replaced is pointed out as "visited" in order to avoid considering it again as a candidate for refinement.

Stopping criterion of the algorithm:
The multi-level partitioning algorithm stops when one of the following conditions is realized:

- a solution is found that is considered "feasible" (solution whose cost is less than the threshold first set by the user);
- no feasible solution has been found, but all the « leaf » entities in the hierarchical tree have been "visited" and marked, and no new decomposition is possible ;
- a maximum number of iterations, first set by the user, has been reached.

It is obvious that, in the first case, partitioning is achieved, since it allows finding a feasible solution, while in the two last cases, it has failed, since it has not been able to determine a solution.

4 Simulations and results of multi-level partitioning

In the case of the benchmark presented above, and since there is no feasible solution (with a positive cost), it is necessary to apply a refinement on the entities that are considered "complex". In PARME, a non-realistic solution is affected a negative cost. Three entities (E0, E2 and E5) present the most important execution times, and thus the highest costs, according to the cost function we chose. These entities are declared "complex" and are subjected to a first refinement. This step gives 10 entities (by replacing the three complex entities by those of the directly below level in the tree).

The application of the exact method to the refined benchmark is still possible, since the number of solutions stays relatively low (1024). It is nevertheless clear that the application of a manual method becomes impossible. The results achieved by the exact method confirm the complexity of the problem: there exist only two feasible (positive cost) solutions to this problem. The first one, at the 381st iteration, gives the following cost: 232299.984375; the second one is the optimum, achieved at the 943rd iteration. It gives the following cost: 213177.

At this point, it is not necessary to apply other refinement iterations, since an optimum was reached. This benchmark is, despite its small size, a problem that is difficult to solve. It has only two feasible solutions and one optimum over the 1024 possible combinations. There is a probability of 0.00097 to get this optimum.

The fact that we knew in advance the optimal solution, by applying the exact method, allows us to study, in the following sections, one of the several heuristics that have been implemented in PARME environment: the genetic algorithm, and its parameters.

Table 1 summarizes some of the results achieved using the genetic algorithm with different strategies and sets of parameters. The selection method used in all cases is "the lottery wheel". The performed simulations show that the solution "0000000000" (which means that all the entities have been mapped on the software processor named "0": the hardware one), that has a cost of –27902, is a local optimum towards which the algorithm often quickly converges, if the parameters are not well chosen. This is due to one of the main drawbacks of genetic algorithms which is the premature convergence, that often leads towards local optima.

The mutation operator tries to correct this problem, but with this benchmark, when this operator is used alone, it appears to be insufficient to get out this local optimum trap (simulations 2 and 3 in table 1). Simulation 3 shows that too small crossover and mutation parameters avoid diversifying the population, and leads to an early convergence towards local optima. Too high values, avoid the algorithm to correctly explore the solution space, by forcing it to change too frequently the search region, and give the result that we can observe in the first simulation of table 1: where the crossover and mutation probabilities are both equal to 1.

Table 1. Genetic algorithm results.

N°	NG	S	A	C	M	EC	It	ET
1	100	1	0.01	1/CRP	1/CRP	-70846	1	1.96
2	50	--	--	0.9/CRP	0.07/CRP	-27902	5	8.97
3	100	1	0.01	0.5/CRP	0.01/CRP	-27902	5	11.92
4	60	1	0.01	0.9/CRP	0.1/CRP	213177	7	12.96
5	100	5	0.01	0.9/CRP	0.09/CRP	213177	8	15.97
6	100	1	0.01	0.9/CRP	0.9/CRP	213177	12	20.95
7	25	1	0.01	0.9/NBS	0.1/NBS	-27902	7	13.94

NG: Number of Generations; **S:** Sigma (**Sharing**); **A:** Alpha (**Sharing**); **C:** Crossover probability and replacement technique; **M:** Mutation probability and replacement technique; **NBS:** N Best Selection (The N best individuals are selected); **CRP:** Children Replace Parents; **EC:** Elite solution Cost; **It:** Iteration number; **ET:** Execution Time (sec.);

Notice that no feasible solution has been reached for populations under the size of 25. The influence of the population increase on the execution time is obvious and easy to understand, the operations on each generation being repeated a greater number of times.

The only way to get out of the local optima is to use the ***sharing*** strategy. Sharing allows reducing the number of identical individuals. But it is not sufficient to diversify the population. This is why the strategy "***children replace parents***" was used for selecting the individuals at each generation. The combination of these two strategies gave the best results and was the only way to get the optimum solution. This is illustrated by simulations 4 to 6 in table 1, where the best solution is achieved in 100 per cent of the cases.

The Iteration Number and Execution time correspond to the iteration at which the best solution is achieved and the population size is always set to 100;

5 Conclusion

The work presented in this paper is a practical case of the mutual cooperation that can take place between technology and natural phenomena. Genetic algorithms are an example of such cooperation, used here to solve the partitioning problem in codesign.

Partitioning at a fixed granularity degree has showed its limits. This paper presents a new algorithm that dynamically modifies the granularity level, by splitting the application entities that are considered as too "complex". Coupled with genetic algo-

rithms, this approach allows finding feasible solutions when no fixed granularity partitioning is able to.

References

1. KUMAR S. AYLOR J.H. JOHNSON B. AND WULF W.A., "The Codesign of Embedded Systems", Kluwer Academic Publishers, 1996.
2. DE MICHELI G. AND GUPTA R., "Hardware/Software Co-design", Proceedings of the IEEE, Vol.85, N°3, pp.349-365, 1997.
3. THEISSINGER M., STRAVERS P. AND VEIT H., "CASTLE: an interactive environment for HW-SW co-design", Proceedings of the Third International Workshop on Hardware/Software Codesign, pp.203-209, 1994.
4. ISMAIL T.B., ABID M., O'BRIEN K. AND JERRAYA A.A., "An approach for hardware-software codesign", Proceedings of the Fifth International Workshop on Rapid System Prototyping, Shortening the Path from Specification to Prototype, pp.73-80, 1994.
5. EDWARDS M.D., "A development system for hardware/software cosynthesis using FPGAs", Second IFIP International Workshop on Hw-Sw Codesign, 1993.
6. D'AMBROSIO J.G. AND HU X., "Configuration-level hardware/software partitioning for real-time embedded systems", Proceedings of the Third International Workshop on Hardware/Software Codesign, pp.34-41, 1994.
7. POP P., ELES P. AND PENG Z., " Scheduling driven partitioning of heterogeneous embedded systems", Dept. of Computer and Information Science, Linköping University, Sweden, 1998.
8. BALARIN F., HSIEH H., JURECSKA A., LAVAGNO L. AND SANGIOVANNI-VINCENTELLI A., "Formal Verification of Embedded Systems based on CFSM Networks", in Proc. DAC, pp.568-571, 1996.
9. KALAVADE A. AND LEE E.A., "The extended partitioning problem: hardware/software mapping and implementation-bin selection", Proceedings of the Sixth International Workshop on Rapid Systems Prototyping, Chapel Hill, NC, June 1995.
10. GAJSKI D.D., NARAYAN S., RAMACHANDRAN L. AND VAHID F., "System design methodologies: aiming at the 100h design cycle", IEEE Trans. on VLSI Systems, Vol.4, N°1, pp.70-82, March 1996.
11. HARTENSTEIN R., BECKER J. AND KRESS R., "Two-level hardware/software partitioning using CoDe-X", IEEE Symposium and Workshop on Engineering of Computer-Based Systems, March 1996.
12. ELES P., PENG Z., KUCHINSKI K. AND DOBOLI A., "System Level Hardware/Software Partitioning Based on Simulated Annealing and Tabu Search", Design Automation for Embedded Systems, Kluwer Academic Publisher, Vol.2, N°1, pp.5-32, 1997.
13. BENATCHBA K., KOUDIL M, DRIAS H., OUMSALEM H. ET CHAOUCHE K., "PARME un environnement pour la résolution du problème Max-Sat", *CARI'02*, Oct. 2002.

e y de g App ed G e dy f R d

d s ld s t d t z S t

D ar a n o Ing n r a a In or ac on a o n cac on
n r a rc a a nar o 00 rc a S a n

A st t So co ng can a ow r a roac n agr c ra o a n ro wo ff r n an n r ng r c r on ng o o a n o o n ron n n or r o cr a an a rox a or o x a n a o o r a on w c r c a concr agr c ra roc con n gra on o n ro o n a G ogra ca In or a on S n w r a a an ora n r o a on a c n ra In a r w r n w a on o o co ng or o n on r c In ar c ar r w o rr ga on wa r n or a concr on o S a n an ow ow n ro o can n gra n a GIS n or r o o c a rox a ow r

n uc i n

S t p t d ls ppl d t sp t l d t p l t p l t t t t t lt l d t s w ppl t s t p t s tw ld st w s t s st d t t s w t d s t t l p ss s t t dd s d l t s t s p d ll s d l l t d t t d s d w ls s t d l d pl t s t t d p t p l t S st S p l t t t s p p t t ds t s st d t t s l t d w t t w t s pt s z s t t d t S t st Sp t t s ff s p t t l t w t lt pl s p t t l t l l t s t s d w t s p t t t s w t p d t t w t ds d p d t p d t t d lt t d t z w t lt t s l t d s l t p p p s t s s t p t t s s p d t d l d pl t s d S s t pp t t t t p t d s t t t d pp t t d t p p w t l w t p l s t t s w lt t p t s t s t w t ds s l tt l tt s s tl t p p s s ll ws s t w t d t p l t st t t w t t t s s t p t

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 401-408, 2003.

s st d t t s tl d s t s d t d t pl t t s t d t w pp t d t s w ll t t d s t w s w d l s s lts t d d ll s t 6 l s s d t w s p t d t

i i n f c cu us f i i n w n s

S s t s st t t s d l p d t t d s l t t s d t t l l t s l p t s p ss l ss t s d t t ld t st d t t s d t s st s d t w s st t t s pp t d t Sp s t s z t s sp d t d ff t z s t t l 6 lt l st t s w p d ll t s s s d d p t s s dw d s p d tw d st t d t t t p p l t z s d ff t s s s t t d l s l t p t d t w d p p t t s l d t t t t s d l t s d t t ll t s t t p llst st t s t S t s ld t s d S l t l t s t S st w s st t s p s t p t t p t ll t d t ll d d s s s s s t s t s l d t s s t t s t 6 st t s t t t l t t d s s s l d t s p t t st t t w t d d ps s ss s p t t t t t t s p l s p t t t ds t ps z s t w ld t t l w l l t w t d d t d t lt t s p s l t s t s t st Sp t p l z w t s w s l t d ll w w w ll t d t p l t st t t w t ds s s s t d d lt z t t U t d t s [2] d t l d pl t s s t s s t *T* s t p p t sp t s s t t l p t t st t p w t d s t s t t w t t t s l st s l p t d pl t t sp t s s p s l t w t t t s t p t t t lt t *T* d p ds t p t l l t d t s d t d p t t s s d t *T* s t d t s ppl d t t d t d st t l t t p t wt st ll w ps d w t t ps st pl t d S t *T* $_{op}$ t s t d s *T* $_{op}$ *T* ll t w t d d l t t d t d w t s t d t d ff *T* $_{op}$ d ff t ll w s t t w t s l t p l p ss s *T* $_{op}$

//www car /cagr/c a/ n x a
//www ao org

T p t s t d s t t d l
t d l s d l z t lt t d ts t w t t
p p s st t t w t ds pl t s s
d l t l ss t d l t t d s t t t d t s
d l s d p t d w d s w l t s d d ls
d l p d t d p d t s ss l t st s d d ls
w d s s t s ll d d t d l t w s p p s d s
d tt tl s t p d d ls t
w s ts s ws t s d l sts p t d t
pp s d t l w t t t d t st t w d s t
t t t s s t ss t t w t t st t t t d s t t st
d st t ll d s d st t d t s d d ll d
st t t s w t s p t s d d t sp d z
t d t t ss t d st t s s t St t s ss t s
s d t w t l t t s p p w p p s t s
s t p t t pp t d t l s t l st t s t t l s
t s l t ll p ts t p

C us in A i s n A f fu y in

w t d ts t ppl t t l t ll s d
t s t lt l t S pl s t s
6 t s s t w p p s t s zz pp t pp t
s l d t s s p t

t t d w t w t s w s t
t 6 t t d t d s
d t s d s l t st t d t d s
t l t d t s tw t d t s t t st t
p d t p t d t t st t d
d t s t p t p d t tp t d t t s d
t t d t d s t d w t t p ss

d t *f* w ll pp t d s t t $\hat{f}$
s t p t st t w w ll s zz d l w s l
p p s d t t w t l st l t s d S
s p p s d st st p s t t l st l t s ppl d t
t p t tp t d t s s w ll w S ds s t l p t s
d t t l d l S t s p t s st t l
t d t t p l t zz tw d s t s p t
t d t s p zz s ts t l s d t s S t t
l st s s d t p d t l s t zz d l t t t
t t ds t d s d s p t t s l t s w ll
d st t t t ds t s
l st s t p t tp t p d t sp s d t p t s
p t d s d s t l p t s S tw

n u f

s t w s p sl s d s t d t l t st t s s
t d t S t s st p s ts t l t s st
p t t l t s t s ll d OU t l t t s ts p ts t st s
ll l t pl s st s w s t t t 6 d
t s st s t sl d t t l t p s t d s d s
t OU t l t tw l t s t t t s t pl s
t s stl w t t l t t t s t pl s OU
S dl w ls t Kt l t t t s t pl s
OU t l s t s d l s s s t t s l ts s d
l d t tl s d t d t t l t
t p d t s s w ll s s w t s t pl s t l
s Kd t s ws t st l t t w s s l t d t w t
d t s t
p ss t t d t s t w s s pl st t d
t pl Kt l t t st t d t t pl t s
l t t st t o d t s d t w t
w t pl w t t ll w lds

d t s d st t d t d t t tp t
st t o t t s p d w ll t d d t s t t pl s
t d t s t w ll s d l t s s t t
t pl s s l w t l s t s
tt t s t s l z t p w d l d t w ll
s d sl t s s w ll p ss st s d l d t s p
l t

M s f i i n

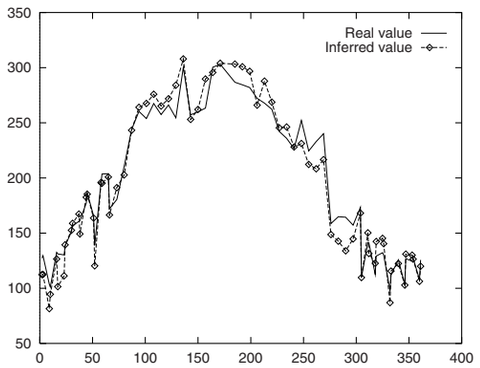
e r r

t t s l t zz d l w s p d t d l
t t S l s ts l S t d S d
w t d ts t t l l z t S s lt d 66 s
d ff tw t tw s s s s t s t d
t st t d l w d w t d l d t w s w d p l t
t st t d t t t d pl t p s w d t l s
pp t d t p t st t s d d t p t t p
d t l t d t s w t sp t t t p t d p t d t S
w t st t s ld s d pp t pp s
t st ll p ss l st t s t l z t
t l s t d s s p t t pl s sp d t d ff t

S a on co	Da a a o n	R S
		2
	0	
J	0	0
J	0	
J	0	
I		
		0
	0	
R	0	**2**

G n ra a on rror ng ac can a o a n a on a n

st t d t st t d t t t l w S s d w
st t s s d s p t d t l st s w st t 6 s s d
s ws t zz d l p t t w p s st
s t s d t S l s 6 st t st t s w
pp t d s s p t

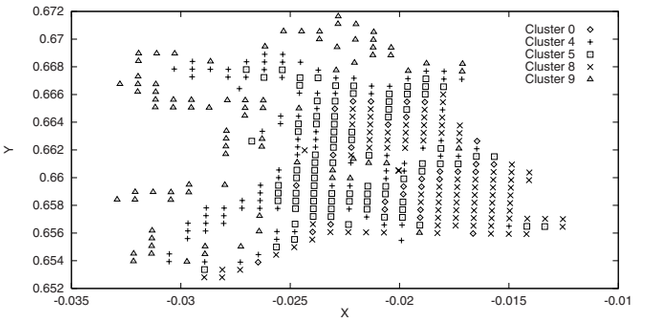


F R a ra a on an ra a on n rr ro R

c e r r

t s t l l d l w s t d t w l s
d d t pp t p t s d t

st t st l t d t s d ff t pp l l d l w t d s lt t l st t w p l s l st s pp d p t d



F 2 G ogra ca c r o rc a r g on ac c r ra a on w r a rox a ng a n ro oca o

d t t d t t t p t stl t s t d d d w t l st t p t l s t d s t d l sp d t t t l st s ppl d t t l w t s w d ls t p d t s lts t l l d l t ll t s d t t s t l l l d t l t l st t t t t p t tt t s t t t l st s t sp d l d l st s s ld l d s st t s s p ss l s t w t s l st s w t st t s w d d t d l p t d t s t t l st s t st t s t s l st s ld d t s t s d t d t s s t t s t t t t w ld d s l t s w l d t s lts l w l t

t l t d st t t st t s t l st s l t s w t t d t s l l d t s t d t d ff t l l d t s ts s w l st s d d t d t l st s d t d t st t s st l st 6

s s lts s w t t l l d ls d t p t l l t l st s d t l l t l st l t t t p t s t l l d l l st t S d l z t S 6

3 r ce

l l s t t t d t s ll t p ts t d t d pl t zz d l t d s d S

Cluster	Stations	Tuples	RMSE train	RMSE test	Rules
0	AL41,CA42,CA72,CA91,MO12 MO22,MO41,MO61,MU31 MU62,TP73,TP81	2454	5.56	58.02	14
1	JU12,JU71,JU81,LO21	-	-	-	-
2	AL71,JU52,LO41	-	-	-	-
3	CR12,CR52	-	-	-	-
4	AL31,AL51,CI22,CI32,CI52,LO11,	1063	5.41	57.62	6
5	ML12,ML21,CA21,CI42,LO51, MO31,MO51,MU52	-	-	-	-
6	-	-	-	-	-
7	CR32,JU42	-	-	-	-
8	AL62,CA52,LO31,MU21,TP22,TP42	-	-	-	-
9	CR42,CR52,JU61,LO61	1057	9.03	22.62	6

Table 2. Clusters found in the geographical space of Murcia region.

As an example, figure 3 shows the radiation surface (the one above) for the whole Murcia region (the below surface) which is compound of 777 different geographical points, for a concrete day of the year. Again, CR12 station was used as input for the whole bunch of points.

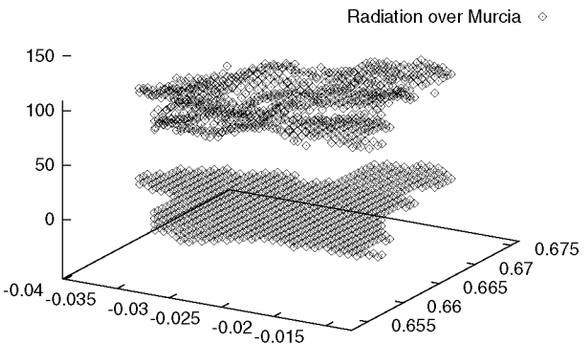


Fig. 3. Radiation inferred for the whole surface of Murcia, for the first day of the year. CR12 station was used as main station for the whole bunch of points (i.e. 777 points).

6 Conclusions and Further Works

In this paper a neuro-fuzzy approach for the approximation of the radiation for a geographical point and a given day starting from the radiation known for

t p t d t l t d d t s s s w s lts t
l l d l s l d t w d s t s lts
s t sp ll l l d ls
p ss l s d t d t t tt pp t d ls
s l t d w t t d t s sp t s sp d t
p l d l t d l t t d d lt t d d t t d
t t d t t p ts t s t st t s t
d s l sp d d t d st t t s t t
l t d lt t d s t s t s p t t t t d t
ll t ld t st t t t t tw d ff t pp s
d s d t s t l d w ps ss t d st t s
d t s l t p l t l d t s t p l t d
t t w t d t t zz l s d ls l l d ff t d
l st s t t w d t p t l t t t l st
st t t p t w d t S t t st p t
zz l s d t d l ss t p

f nc s

ro on on So co ng con rg nc o rg ng r a on ng c
n q *na S ng*
J o a Go S ar a a a G Sanc an a r c ng
rr ga on wa r n ng o co ng c n q In *c ng*
c n nc n Sy c yb n c an n a c o III o r
Sc nc an ng n r ng ar II ag 0 r an o or a S J 000
J an o a S ar a a an a a Da a n ng a o
rr ga on wa r anag n In *c ng* *1* Grana a S a n
J n 00 o a ar
ro w r an o *ga n a anag n ga n a*
oo an gr c r rgan a on o n Na on ra n
n ng an a //www ao org/ ocr /S 0 / 0 00 a a a a
on n on Jan ar r 00
S o n ca on a on c r a on *na*
n g n an zzy Sy
R Ing an G row N ra n wor o or r c ng organ c a r
con n n a a c wan o *ana an Sy ng n ng* 00
J S Jang n a n wor a n r nc *an n*
Sy an an yb n c 0
ng *Sy n ca n y* r n c a
S on an G ran r r c a N ra N wor n gr c ra ac n
r ca on In *S a* cago I no S

Evolutionary Multi-Model Estimators for ARMA System Modeling and Time Series Prediction

Grigorios Beligiannis[1,2], Spiridon Likothanassis[1,2], and Lambros Skarlas[1,2]

[1] Department of Computer Engineering & Informatics, University of Patras, GR-26500, Patras, Greece
{beligian, skarlas}@ceid.upatras.gr
[2] Computer Technology Institute (C.T.I.), Kolokotroni 3, GR-26221 Patras, Greece
likothan@cti.gr

Abstract. In this work, a well-tested evolutionary method for system modeling and time series prediction is presented. The method combines the effectiveness of adaptive multi model partitioning filters and GAs' robustness. Specifically, the a posteriori probability that a specific model, of a bank of the conditional models, is the true model can be used as fitness function for the GA. In this way, the algorithm identifies the true model even in the case where it is not included in the filters' bank and is able to accurately forecast the short-term evolution of the system. The method is not restricted to the Gaussian case; it is computationally efficient and is applicable to on-line/adaptive system modeling and time series prediction.

1 Introduction

Selecting the correct order and estimating the parameters of a system model is a fundamental issue in linear time series prediction and system identification. The problem of fitting an ARMA model to a given time series has attracted much attention because it arises in a large variety of applications, such as time series prediction in economic and biomedical data, adaptive control, speech analysis and synthesis, radar, sonar, etc. Furthermore, the effect of varying process parameters on system performance is of crucial importance, specifically when the desired system performance alters and deteriorates due to the disturbances of the system's parameters, whenever the design is based on nominal parameter values. Therefore, it is more than desirable to be able to incorporate parameters' variation into system design.

There exist various methods for system model identification and time series prediction that represent information theoretic criteria. The most well-known of the proposed solutions include Akaike's Information Criterion (AIC) [1], the Final Prediction Error (FPE) and the Minimum Description Length (MDL) Criterion [10]. Most of the techniques that are computed by the above criteria assume that the data are Gaussian, are based upon asymptotic results and are two-pass methods. Therefore, they cannot be used in an on-line or adaptive fashion. A different adaptive approach, based on the Partitioning Theorem, is the Multi Model Adaptive Filter

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 409-416, 2003.

(MMAF) [11] that operates on general, not necessarily Gaussian data pdf's. The MMAF converges to the optimal solution, if the model supporting the data is included to the initial filter's bank. Otherwise, it converges to the closer model by mean of the Kullback information criterion minimization. This is due to the fact that the number of filters used in the MMAF bank is finite.

Genetic Algorithms (GAs) are known to be one of the best methods for searching and optimization [8], [9]. By applying genetic operators (reproduction, crossover and mutation) in a population of individuals (sets of unknown parameters properly coded), they achieve the optimum value of the fitness function which corresponds to the most suitable solution. As a result, they converge to the (near) optimal solution by evolving the best individuals in each generation. The main advantage of the GAs is that they use the parameter's values instead of the parameters themselves. In this way they search the whole parameter space.

The paper is organized as follows. In section 2 the ARMA model system identification problem is stated. In subsections 2.1 and 2.2 the proposed method and the GA structure are presented respectively. In subsection 2.3 the computational aspects are discussed Section 3 contains the simulation results, while section 4 summarizes the conclusions.

2 ARMA Model Order Identification

As known, a general model for ARMA can be represented as follows:

$$y(t) \quad \sum_{i\ 1}^{p} a_i y\ t \quad i \quad \sum_{j\ 1}^{q} b_j e\ t \quad j \quad e\ t \tag{1}$$

where y(t) is the observed data, which is supposed to be stationary, e(t) is a zero-mean white noise process, not necessarily Gaussian, with variance R, which denotes the series of errors, n=(p,q) is the order of the predictor and a_i (i=1,...,p), b_j (j=1,...,q) are the predictor coefficients. Clearly the problem is two-fold: one has first to select the order of the predictor and then to compute the predictor coefficients. Of course the most crucial part of the problem is the former.

First of all, the problem has to be reformulated in the state-space form [2]. After that, let us assume that the order n, or the parameters (p,q), of an ARMA model is unknown and the only available knowledge about the true order is that it satisfies the condition $p_0 \le p \le p_{MAX}$ and $q_0 \le q \le q_{MAX}$. It is clear then that the true model is specified by the true value of parameters (p,q). The problem is then to select the correct model among various candidate models. In other words, we have to design an optimal estimator (in the minimum variance sense), when uncertainty is incorporated in the signal model. The solution to this problem has been given by the multi model partitioning theory [6], [7].

2.1 ARMA Model Order Estimation Using Evolutionary MMAF

In the ARMA system model identification problem there are two quantities, which must be estimated; the model order n and the unknown predictor coefficients a_i,b_j. The estimation method is the MMAF as it is obtained from Lainiotis partition theorem [6], [7]. Our goal is to achieve the optimal estimation for both estimated quantities and particularly for the model order n=(p,q). The only information we have for these parameters is that they belong to a set or a space (finite or infinite). It is obvious that if the unknown parameters belong to a finite-discrete set with small cardinality, the MMAF is the most appropriate and effective method to estimate them. It is also widely known that EAs perform better when the space, which will be searched, has a large number of elements. So, EAs can be used when the unknown parameters belong to a space with large cardinality or belong to an infinite space or follow a probability distribution. Then, we should optimize with EAs the model-conditional pdf. That means that we have to optimize, for discrete sample space, the following probability function [4]:

$$p\ n/t \quad \frac{L\ t/t;n}{\sum_{n} L\ t/t;n\ p\ n/t \quad 1}\ p\ n/t \quad 1 \tag{2}$$

which will be the fitness function for the GA, for the several values of the unknown parameters n=(p,q) and a_i,b_j underlying to the above constraints. The structure of the system of the proposed solution is shown in Figure 1, assuming that the number of the Kalman filters of the MMAF equals m.

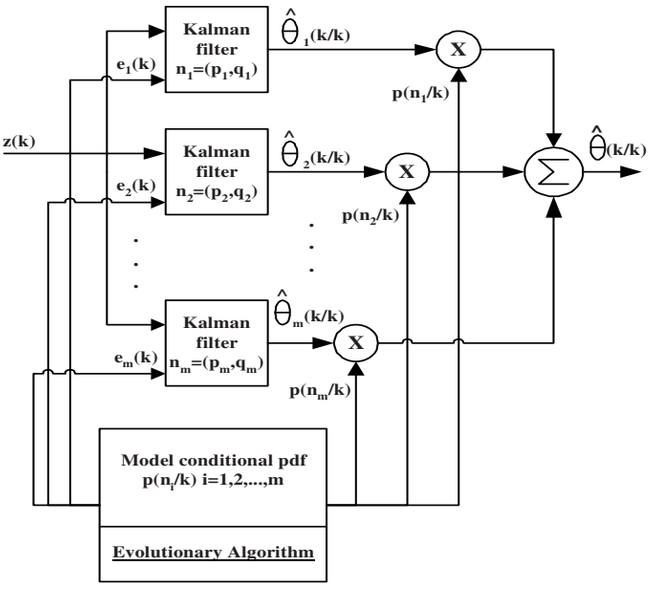


Fig. 1. The structure of the evolutionary MMAF system

2.2 The Structure of the Evolutionary Approach

The structure of the Evolutionary Program (EP) that has been developed is shown in Fig.2. We first make an initial population of s vectors of m pairs of integers (each pair representing a possible value of the ARMA model order). For each such vector we apply a MMAF and have as result the model-conditional pdf of each value. The biggest pdf is the fitness of each vector.

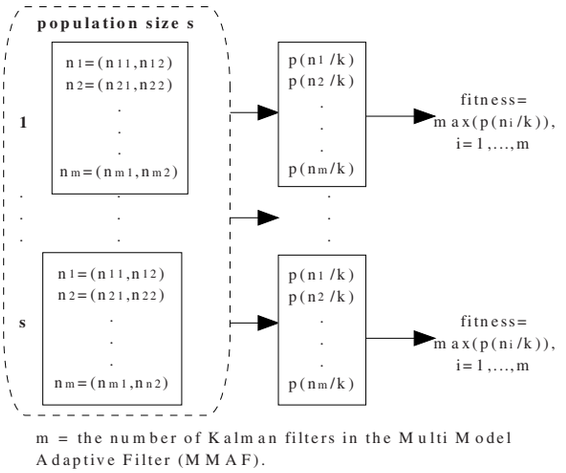


Fig. 2. The structure of the EP

Since we have the fitness of each vector of possible solutions) we are able to perform the other genetic operators, i.e. reproduction, crossover and mutation as stated in [4]. Every new generation of vectors of possible solutions iterates the same process as the old ones and all this process may be repeated as many generations as we desire or till the fitness function has value 1 (one) which is the maximum value it is able to have as a probability [3].

2.3 Computational Aspects

A first look at the structure of the EP, given in Fig. 2, related with the structure of the MMAF, given in Fig. 1, appears very complicated. Concerning the complexity, two aspects have to be discussed. First the time complexity of the MMAF and second the complexity of the EP. Fortunately, the structure of the MMAF although complicated, is by nature parallel. So, in parallel implementation, the computational burden equals to that of one Kalman filter plus the computations required for the computation of the a posteriori probabilities. Furthermore, it has been proved in [5] that the MMAF converges faster than the simple KF. So, the total time required for the MMAF to converge, is practically less than that of the KF. However, one is faced

with a serious design question, concerning the cardinality of the finite subset of parameters to be chosen. The computational burden of the MMAF is proportional to the number of filters it employs; hence, a subset of large cardinality leads to an increased computational load, which may even make the implementation of the searching the whole parameter space, by evolving a population with a small number of individuals. Furthermore, in system theory, usually we are interested in adaptive systems, by mean of sensing the changes of the systems parameters.

Although, the GAs were inspired from the adaptation process in nature, their implementation is not adaptive, in the sense of the on-line adaptation, since they converge slowly. The new generation of the parallel machines enables the parallel implementation of the GAs, since they work on populations of individuals, thus reducing the computation time. In order to implement an adaptive scheme of the proposed EP, we can use the following heuristic. First, we implement the MMAF by using the best solution of the initial population and we continue the execution of the EP separately. Then we can update the system's parameters, after a sufficient number of iterations, or when a significant increase of the performance has been occurred. This scheme works sufficiently, since the performance of the EP increases dramatically after a small number of generations. Finally, the increase of the system's dimension does not affect significantly the computation time, since instead of low dimension vectors, the system has to manipulate large vectors or matrices. In this case, one can apply second order parallelism [5], in the level of matrix manipulation, thus saving time. Furthermore, in the case of sequential implementation, as it has performed in [4], where a similar problem for multivariate system structure identification is presented, although the computational effort has been increased significantly, the algorithm converges very fast.

3 Experimental Results

The presented algorithm has been run extensively on several system identification and time series prediction experiments. All of them were carried out 100 times (100 Monte Carlo runs). The size of the GA's population we used in all our experiments was 10, the crossover probability was 0.95 and the mutation probability was 0.15. Also, the number of Kalman filters in the MMAF was 10 and for every generation of the GA the MMAF was applied for 35 runs.

3.1 ARMA System Identification (Modeling) Results

In this subsection, an ARMA system identification (modeling) problem is discussed, where the process model changes during the operation and is given by:

a) for the first 400 samples at time instant t=0:

$$y(t)=1.8y(t-1)-0.9y(t-2)+0.4e(t-1)-0.8e(t-2)+e(t)$$

b) for the last 600 samples at time instant t=400:

$$y(t)=1.5y(t-1)-1.4y(t-2)+0.8y(t-3)+0.7e(t-1)-1.1e(t-2)-2.3e(t-3)+1.5e(t-4)+e(t)$$

The proposed technique succeeded in identifying the correct model order and estimating the true values of the unknown parameters as shown in the figure bellow.

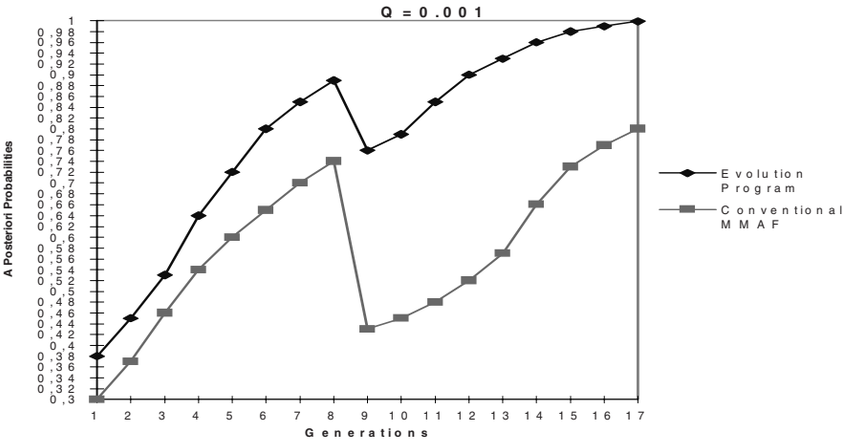


Fig. 3. The evolution of the A-Posteriori Probabilities of the best genome compared to the conventional MMAF.

3.2 Time Series Prediction Results

In this subsection, the proposed technique is applied in order to predict the evolution of a time series consisting of economic data taken from the Stock-Exchange House of Athens. The application of the algorithm results in estimating an adaptive ARMA model able to accurately forecast the short-term evolution of the system. The time series used consists of the High, Low, Start, Close and Volume values for two different main stocks – the stock of The National Bank of Greece and the stock of the National Communication Organization of Greece – from 1/1/1993 till 31/12/2002. The proposed technique succeeded in predicting accurately the evolution of the stock values as shown in the figures bellow. In these figures the comparison between the true values and the predicted by the program values for both stocks is presented. The ARMA model, estimated by the evolutionary program, resulted in accurately forecasting the short-term evolution of the system, based on the past values of the time series, as proved by the small Mean Square Error (MSE) estimated.

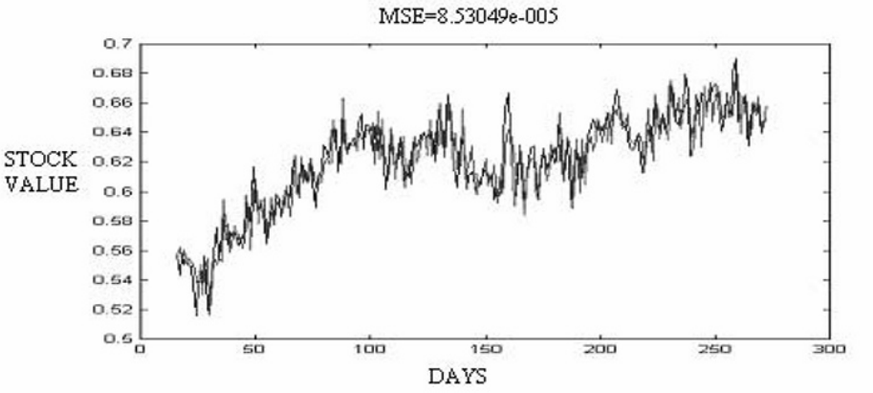


Fig. 4. Comparison of the true values and the predicted values for the stock of the National Bank of Greece

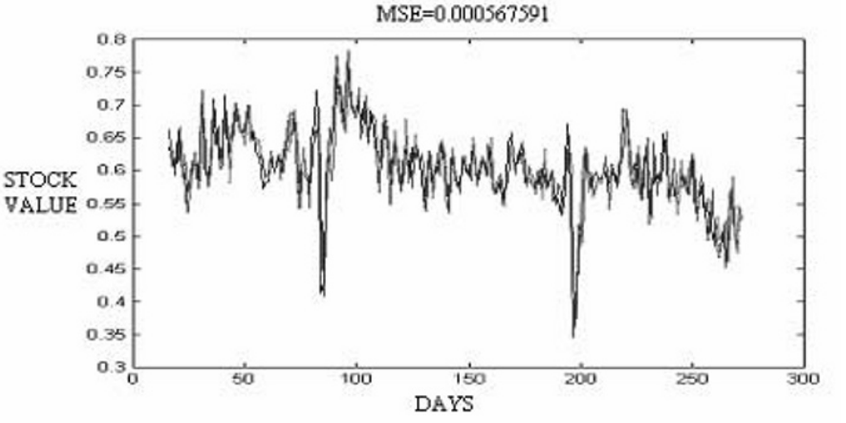


Fig. 5. Comparison of the true values and the predicted values for the stock of the National Telecommunication Organization of Greece

4 Conclusions

In this work, a well-tested evolutionary method for adaptive estimation of ARMA discrete time systems, with unknown order and parameters, has been proposed. The method combines the well known Adaptive Multi Model Partitioning theory with the effectiveness of the GAs. A specific adaptive implementation that alleviates the disadvantage of the time consuming implementation of the GA is proposed. Simulation results showed (Fig.3) that the proposed hybrid evolutionary method performed better than the conventional MMAF and the other conventional methods since it has the ability to search the whole parameter space and track successfully every change that may occur in the order of the system model. The superiority of the algorithm is obvi-

ous especially in cases where the true order of the model suddenly changes. Most of the conventional methods either were not able to track the sudden change or took them too long to do it. As far as the presence of local minima is concerned, the proposed hybrid method succeeds in alleviating this problem by using a quite high mutation probability throughout the whole identification procedure. What is really remarkable is that the proposed method tracks very quickly the sudden changes between local and global minima and as a result its transient performance does not affect its overall performance. Although the parameters' coding is more complicated a variety of defined crossover and mutation operators was investigated, resulting in accelerations of the algorithms convergence. Furthermore, the evolution of the initial population results to the identification of the true model, even in the case where it is not included in the initial population of the filter's bank. Except for that, the method performs significantly better compared to the pure evolutionary approaches. In contrary to the evolutionary ones the proposed method is able to cope with the complexity of the model and reliably lead to the correct order and the true parameter's values. Finally, the method can be implemented in a parallel environment, since the MMAF as well as the GA are naturally parallel structured, thus increasing the computational speed.

References

1. Akaike H.: A New Look at the Statistical Model Identification, IEEE Trans. Automat. Contr, vol.26, (1977), 1-18
2. Beligiannis G. N., Demiris E. N. and Likothanassis S. D.: Evolutionary ARMA Model Identification with Unknown Process Order, 3rd IMACS/IEEE International Conference on Circuits, Systems and Computers (CSCC'99), Athens, Greece, July 4-8, (1999)
3. Berketis K. G., Katsikas S. K. & Likothanassis S. D.: Multimodel Partitioning Filters and Genetic Algorithms, J. of Non-Linear Analysis, vol.30-4, (1997), 2421-2428
4. Katsikas S. K., Likothanassis S. D., Beligiannis G. N., Berketis K. G. and Fotakis D. A.: Evolutionary Multimodel Partitioning Filters: a Unified Framework, IEEE Transactions on Signal Processing, vol.49, No.10, (2001), 2253-2261
5. Katsikas S. K., Likothanassis S. D. and Lainiotis D. G.: On the Parallel Implementation of the Kalman and the Lainiotis Filters and their Efficiency, Signal Processing, vol.18, (1991), 289-305
6. Lainiotis D. G.: Partitioning: A Unifying Framework for Adaptive Systems II: Estimation, Proc. IEEE, vol.64, (1976), 1182-1198
7. Lainiotis D. G.: Partitioning: A Unifying Framework for Adaptive Systems I: Estimation, Proc. IEEE, vol.64, (1976), 1126-11436.
8. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. 3rd edn. Springer-Verlag, Berlin Heidelberg New York (1996)
9. Mitchell M.: An Introduction to Genetic Algorithms, A Bradford Book, The MIT Press, Cambridge, Massachusetts, London, England, (1995)
10. Rissanen J.: Modeling by Shortest Data Description, Automatica, vol. 14, (1978), 465-471
11. Watanabe K.: Adaptive Estimation and Control. Partitioning Approach, Prentice-Hall, U.K. (1992)

Real-Coded GA for Parameter Optimization in Short-Term Load Forecasting

Hara P. Satpathy

Barry University
11300 N.E. Second Avenue, Miami Shores, Florida 33161

Abstract. This work explores the possibility of real-coded genetic optimization of the parameters of a Fuzzy Neural Network (FNN) for short-term electric load forecasting. A new approach is proposed to optimize the connecting weights and the structure of membership function. The adjustable parameters of the network such as connecting weights and rule sets of the network are viewed as constraints that we impose on GA in the learning process. Instead of working on the conventional bit by bit operation, both the crossover and mutation operators are real-value handled. The effectiveness of the proposed algorithm is demonstrated by comparison to a non-GA based forecasting approach. The algorithm is comprehensively tested with actual load data of an electric utility and the Mean Absolute of Percentage of Error (MAPE) is below 2.0% in most.

1 Introduction

The fundamental approach to optimization is to formulate a single standard of measurement, a cost function, that summarizes the performance or value of a decision and iteritavely improves this performance by selecting among the available alternatives. Most classical methods of optimization generate a deterministic sequence of trial solutions based on the gradient or higher order statistics of the cost function. These techniques can be shown to generate sequences that asymptotically converge to locally optimal solutions, and in certain cases they converge exponentially fast. However, the methods often fail to perform adequately when random perturbations are imposed on the cost function. Further, locally optimal solutions often prove insufficient for real world problems.

It is frequently observed that when implementing a particular network, various parameter settings are left for us to decide. The performance of a standard backpropagation (BP) algorithm based network depends on the initial choice of the connecting weights, the learning rate and the momentum factor. A poor parameter setting will slow down the convergence rate and may also result in an incorrect output from the model. This necessitates optimization of the network parameters for a better performance. Similarly, in fuzzy logic the generation of membership function has traditionally been a task done either iteratively, by trial-and-error or by human experts. This may sometimes result in poor network performance. Such a task as this is therefore, a natural candidate solution for implementing optimization techniques.

Recently, a global optimization technique known as Genetic Algorithm (GA) has become a candidate for many optimization applications due to its flexibility and

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 417-424, 2003.

efficiency. GA is a searching algorithm [1], [2], [3] which uses stochastic operators instead of deterministic rules to search for a solution. GA randomly searches from point to point which allows it to escape from local optimum in which other algorithms might lag. Another advantage of GA is that it searches for many optimum points in parallel. Due to these advantages, GA's have been successfully applied to many diverse areas such as function optimization, system identification, control system, image processing, combinatorial problems, ANN topology determination and rule based systems.

Most of the GA approaches in use represent the constraint variable using binary form of coding. However, the necessity of using binary coding has received considerable criticism [4] as it is not always the best form to represent the constraint variable. One of the major disadvantages of using the binary form of coding is the slow speed of convergence of the cost function.

Binary coding is again not at all efficient in using the computer memory. The problem existing in binary coding lies in the fact that though only a few bits are actually involved in the crossover and mutation operations, a long string always occupies the computer memory. Therefore, to overcome the inefficient occupation of computer memory, real valued crossover and mutation algorithms can be of a lot of help. This is particularly the case when a lot of parameters are needed to be adjusted in the same problem and a higher precision is required for the final result.

The use of binary strings is now not universally accepted in GA literature. Michalewicz indicates that for real valued numerical optimization problems, floating point representations outperform binary representations because they are more consistent, more precise and lead to faster execution [5].

This study investigates the use of real-coded GA in the design and implementation of a Fuzzy Neural Network called GAFNN, for electric load forecasting and compares it with a simple FNN. In ordinary FNNs the training of the network is mostly done by iterative approach using backpropogation algorithm. The network modeling in the simple FNN normally starts with random set of weights and an arbitrary number of fuzzy sets which makes the network susceptible to poor performance because of the improper selection of initial parameters. We therefore formulate the problem of finding the optimum number of rules of the FNN system under consideration; as a constraint of the GA in searching for the optimized rule set. The GA learning paradigm is very powerful since it requires no prior knowledge about the system's behavior in order to formulate a set of rules through learning. We also devise a GA based adaptive mechanism for weight updation. The network is trained and tested using a typical load-data over a two year period.

2 Overview of the Proposed Approach

2.1 Overview of GA

GAs work with a set of artificial elements called a *population*. An individual (string) is referred to as a chromosome, and a single bit in a string is called a *gene*. GAs generate a new population (called *offsprings*) by applying the genetic operators to the chromosomes in the old population (called *parents*). Each iterations of genetic

operations is referred to as a generation. A fitness function, i.e. the function to be maximized, is used to evaluate the *fitness* of an individual. One of the important purposes of GAs is to reserve the better *schemata*, i.e. the patterns of certain genes, so that the offsprings may yield higher fitness than their parents. Consequently, the value of fitness function increases through each generation.

The strings are generated randomly and evaluated using a fitness function. *Crossover* operator starts by picking pairs of strings from the population. Random positions in the string are then chosen and the selected segments are swapped with the other string, similarly partitioned. *Crossover* promotes exploration of new regions in the search space. It provides a structured, yet randomized mechanism of exchanging information between strings. *Mutation* is treated as the secondary operator with the role of restoring the lost genetic material and ensures that the probability of searching any region in problem space is never zero.

2.2 Overview of the Fuzzy System

One of the salient aspects of Fuzzy Inference System (FIS) is the determination of the knowledge base (KB) which consists of the mechanism for developing membership functions, fuzzy reasoning mechanism, and finding the number of rules and the rule base. The network consisting of *input, fuzzification, inferencing, defuzzification* layers is shown in figure 1. There are *N* number of input variables with *N* neurons in the input layer and *R* number of rules with *R* neurons for inferencing. Thus number of neurons in the fuzzification layer is *N*X*R*.

In our work we have opted to use GA instead of the usual iterative screening mechanism for finding the number of rule nodes *R* and the initial weights of the fuzzy neural network. To illustrate the approach of learning of fuzzy membership function we use a simple fuzzy neural network based electric load forecaster as an example.

The inputs to the model are chosen as follows :

$$X = \left[Y_a(k-\Delta+1)...Y_a(k-n\Delta+1)...\ \ T_a(k+1)...T_a(k-\Delta+1)...T_a(k-n\Delta+1)\right].$$

and, the output $Y = Y_d(k)$.

where, $Y_d(k)$ represents the load
$T_a(k)$ is the temperature at k^{th} hour and
Δ is the time step ahead which forecasting is desired.

The past loads are taken to improve upon the prediction capabilities; the notion being similar to that of auto-regression [6]. The temperatures are included to reflect weather sensitivity of the load. It should be noted that $(k+1)^{th}$ element of the input vector is the *a priori* information of the temperature of the hour at which load forecasting is to be done.

Input to the fuzzification layer is a weighted version of input variables, which can be represented lexicographically as,

$$\overline{\eta} = W_1^T X + W_0 . \tag{1}$$

The set of weights between the input and fuzzification layer are given by :

$$W = \{W_o, W_1\} = \left\{\{w_{ij0}, w_{ij1}\} : i = 1,\ldots,N; j = 1,\ldots,R\right\} \quad (2)$$

alternately, the individual elements of the fuzzification(2^{nd}) layer is given by,

$$\eta_{ij} = w_{ij1}x_i + w_{ij0}\,. \quad (3)$$

Output of each neuron in the *fuzzification layer* is a fuzzy membership corresponding to a particular input variables. The activation(membership) function used for this fuzzification layer is given by,

$$f(\eta_{ij}) = \exp(-|\eta_{ij}|^{\gamma_{ij}})\,. \quad (4)$$

where, γ_{ij} is in the range of $1.0 \le \gamma_{ij} \le 5.0$. During training γ_{ij} is updated in this range, which are initially set at 2 . The output of the fuzzification layer can be expressed as,

$$\mu_{ij} = f(\eta_{ij}) = \exp\left(-|w_{ij1}x_i + w_{ij0}|^{\gamma_{ij}}\right). \quad (5)$$

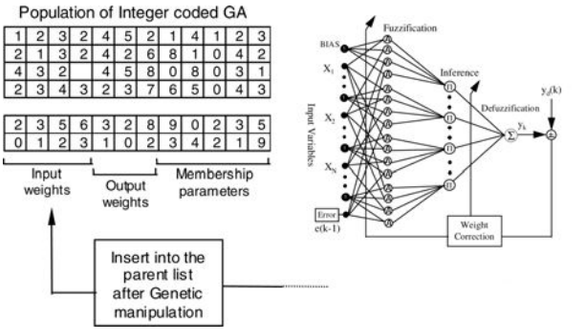


Fig. 1. Forecasting Model

where, μ_{ij} is the value of the fuzzy membership function of the i^{th} *input variable* corresponding to j^{th} rule. The activation function in the *inference layer* (3^{rd}) uses multiplicative inference. The output of this layer is given by ,

$$\mu_j(x_1, x_2, .., x_N) = \prod_i^N \mu_{ij}(x_i)\,. \quad (6)$$

The *defuzzification layer* (4^{th}) has the connecting weights (v_j) to the output from the inferencing layer, and these weights signify the strength of each rule in the output of the model. The output X_{out}, is given as,

$$X_{out}(x_1, x_2, .., x_N) = \sum_{j=1}^{R} v_j \mu_j(x_1, x_2, .., x_N) = \sum_{j=1}^{R} v_j \prod_i^N \exp(-|w_{ij1}x_i + w_{ij0}|^{\gamma_{ij}})\,. \quad (7)$$

3 Training and Model Adjustment

3.1 Training Procedures

A. Iterative Approach using Backpropagation

The network is trained to minimize an objective function. The performance index (*PI*) to be minimized is the *Mean-Square-Error* (MSE) given by, $PI = (y_d(k) - y(k))^2$ where, $y_d(k)$ is the desired output and $y(k)$ is the model output. The model parameters of the proposed fuzzy inference system are updated using the notion of error back-propagation.

We try to optimize the performance index $PI = (y_d(k) - y(k))^2$. The W_{ij1}, W_{ij0} and γ_{ij} are updated till some stopping criterion is reached. The change in W_{ij1}, W_{ij0} and γ_{ij} are computed by differentiating the *PI* w.r.t. required parameters. We define,

$$\varepsilon_k^2 = (y_d(k) - y(k))^2, \tag{8}$$

The training is continued until the following stopping criteria is reached.

$$\sum_{k=i}^{i-50} PI_k \le \varepsilon \quad \text{or,} \quad k \ge I_{\max}.$$

where, $\varepsilon > 0$ and I_{max} is the maximum no. of iterations allowed. In our present implementation I_{max}=2400.

If after completion of training the performance of the model is not found to be satisfactory then, the weights are reinitialized, the number of rules are increased and the above mentioned procedure is repeated. The process is heuristic in nature and therefore takes time to implement.

After achieving a suitable level of performance over the entire training range the *rule nodes* which produce an output close to *zero* over the entire training set, are searched and omitted. Similarly, the inputs for which the fuzzy membership is unity or close to unity over the entire training set are omitted from the network. This is done to optimize the size of the network.

The model is tested without any retraining and if the performance is still satisfactory without the above mentioned *rule-nodes* and *inputs,* then we obtain a model with reduced size. This makes the network computationally more efficient. However, if the performance of the network is not found suitable then the network has to be retrained with the reduced set of rules and inputs.

B. Genetic Algorithm (GA) approach

In the proposed fuzzy neural network in order to optimize the network topology we attempted GA to find the optimum number of rules which gave the best result. A new type of coding called real number coding for the GA implementation was used, alongwith roulette wheel selection mechanism, two point crossover and mutation.

An initial population set of 60 strings was generated randomly and the fitness function was taken to be $1/(1+\eta\varepsilon)$. Here, the *error* ε is given by $\varepsilon = (y_d(k) - y(k))$ with $y_d(k)$ being the desired output and $y(k)$ is the model output of the FNN. The constant η is a problem specific value.

A crossover probability of 0.6 and a mutation rate of 0.005 was fixed. A two point crossover was done as researchers now generally agree that it is superior to the one point crossover method for it can generate more building blocks. The program was run for 5000 generations and the end of it the solution string was decoded to get the optimum number of rules which was found to be 17. Similarly the weights of the network as well as the membership function were encoded together into one string and their optimized values were obtained.

3.2 Simulation Results and Evaluation

The average fitness curve of the genetic optimization of rules is shown in figure 2. Here we find that the fitness value stabilizes to a value of 17 after 25 generations. Similarly, the fitness values for the network weights along with the membership values is given in figure 3. The fitness function attains a stable value after 2000 generations. Figure 4 shows the optimization process of the individual weights of the output layer. Similarly figure 5 shows the stabilization process of some of the membership parameters.

For identifying the parameters of the load mode during network training, weekday data is treated separately from the weekend data. The load curve on Sundays and public holidays is similar in nature, with small deviations. Important holidays like Christmas, New Year day are therefore, treated as Sundays. Other holidays, including Good Friday, Memorial Day, Independence Day, Labour Day and the days preceding Christmas and New Years day are treated as Saturdays. Similar load patterns are used during the training of the network to obtain a faster convergence. As the weekends are treated separately, the weight vector obtained after the Friday forecast is used to predict the load on Monday.

The mean absolute percentage error (MAPE) is used to test the performance of the model and is defined as follows :

$$MAPE = \left(1/N\right)\sum_{i=1}^{N} [\,|\text{forecast load} - \text{actual load}| \text{ x } 100] \,/\, \text{actual load}$$

where, N is the number of patterns in the data set used to evaluate the forecasting capability of the model.

The standard deviation (SD) of the absolute relative error is as follows :

$$SD = \left[\frac{1}{T} \sum_{t=t_0}^{T+t_0} (\%err_t - MAPE)^2 \right]^{1/2}$$

where, T is the number of time samples, and the MAPE is computed over the same range of t. The err_t is the absolute error relative to the peak load for the day.

The special holiday data occurring in the past and the weekend data are used to train the model before the prediction for the holiday is attempted. By collecting similar days from the past, we ensure that the characteristic of only that type of holiday is reflected in the data set.

The performance of the proposed model have been compared with some existing neural tools and time-series approaches [7], [8], [9], [10], [11] as well as with simple

fuzzy neural networks. It has been found that our method performs better than the above models. To evaluate the performance of the proposed network architecture, it is used to forecast one-day ahead Peak and Average load. In the simulations, the data from an utility in the state of Virginia, USA are used. A mathematical software, MATLAB is used in the implementation.

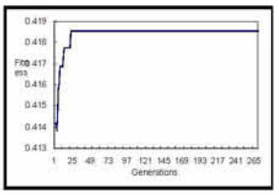


Fig. 2. Fitness values of rules using GA

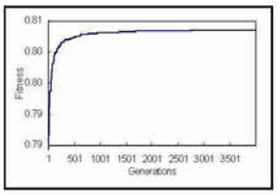


Fig. 3. Fitness values of FNN parameters using GA

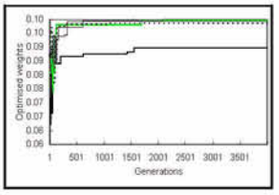


Fig. 4. Optimization process of output layer weights

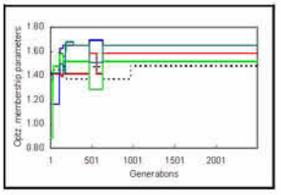
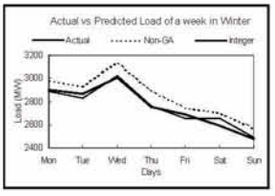
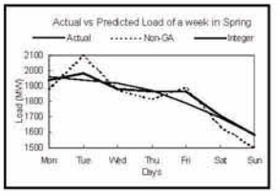


Fig. 5. Optimization process of membership parameters

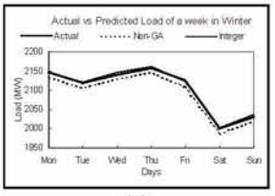


(a)

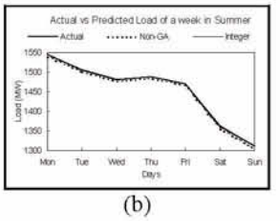


(b)

Fig.6. 24-hours ahead Actual vs Predicted Peak load, (a) Winter (b) Spring



(a)



(b)

Fig. 7. 24-hours ahead Actual vs Predicted Average load; (a) Winter, (b) Summer

4 Load Forecasting Results

The 24-hours ahead peak load prediction over a week in winter, spring, summer and fall are shown in Fig. 6(a)-6(b). It is observed that the predicted output closely follows the actual load pattern and the error was found to be around 2%.The 24-hours average load forecast over a week for winter and summer are shown in Fig. 7(a) and 7(b). It is observed that the average load forecast is better than that of peak load forecast. This phenomenon is due to the averaging effect of the load pattern.

5 Conclusion

In this paper we address the problem of load forecasting using a real-coded genetically optimized fuzzy neural network structure, which gave a significant improvement in performance. The weights in different layers of the network are optimized using a genetic algorithm based approach. Thus the usual problem of random initialization of initial weights in case ordinary fuzzy neural network is avoided. The efficacy of the network is validated by selecting a practical load pattern of all seasons. As summer load pattern is considerably temperature sensitive; the peak load forecast demonstrates the network's efficiency. The weekly MAPE value is mostly within 0.5-1.5% for 24-hours ahead peak load forecast and within 0.05-0.15% for average load forecast which is a very good result in comparison those obtained by a simple fuzzy neural networks.

References

[1] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: Univ. of Michigan Press, 1975.

[2] K. A. De Jong, "On using genetic algorithms to search program spaces", *In Proc. 2nd Int. Conf. on Genetic Algorithms and their applications*, Hillsdale, NL: Lawrence Erlbaum, 1987, pp. 210-216.

[3] D. E. Goldberg, Genetic Algorithms in search, Optimization and Machine Learning. Reading, MA: Addison-Wesley, 1989.

[4] Y. Huang, C. Huang, "Real-valued genetic algorithms for fuzzy grey prediction system", *Fuzzy Sets and Systems*, 87 (1997) pp. 265-276.

[5] Z. Michalewicz, "Genetic Algorithms + Data Structures = Evolution Programs", *New York; Springer-Verlag*, 1992.

[6] A.M. Schneider, T. Takenawa, D.A. Schiffman, "24-hour electric utility load

[7] forecasting", *chapter-7, pp. 87-108, Comparative models for electric load forecasting, John Wiley & Sons Ltd.,* 1985.

[8] O. Mohammed, D. Park, R. Marchant, T. Dinh, C. Tong et al, "Practical experiences with an adaptive neural network short-term load forecasting", *IEEE Trans. PWRS,* vol. 10, no. 1, Feb. 1995, pp. 254-265.

[9] Khotanzad, R. Hwang, A. Abaye, D. Maratukulam, "An adaptive modular artificial neural network hourly load forecaster and its implementation at electric utilities", *IEEE Trans. PWRS*, vol. 10, no. 3, Aug. 1995, pp. 1716-1722.

[10] P.K. Dash, A.C. Liew, S. Rahman, "Peak load forecasting using a fuzzy neural network", *Electic Power System Research* 32 (1995) pp. 19-23.

[11] A.G. Bakirtzis, J.B. Theocharis, "Short term load forecasting using fuzzy neural networks", *IEEE Trans. PWRS*, vol. 10, no. 3, Aug. 1995, pp. 1518-1524.

[12] P.K. Dash, H.P. Satpathy, S. Rahman, "Short term daily average and peak load predictions using a hybrid intelligent approach", *IEEE, EMPD-95, Singapore, Catalogue No. 95TH8130.IEEE*, vol. 2, Aug. 1995, pp. 565-570.

Parallel Computation of an Adaptive Optimal RBF Network Predictor

M. Salmerón, J. Ortega, C.G. Puntonet, and M. Damas

Department of Computer Architecture and Computer Technology
University of Granada. ETS Ingeniería Informática
Campus Aynadamar s/n. E-18071 GRANADA (Spain)

Abstract In this paper we analyze parallel processing in clusters of computers of an improved prediction method based on RBF neural networks and matrix decomposition techniques (SVD and QR-cp). Parallel processing is required because of the extensive computation found in sucn an hybrid prediction technique, the reward being better prediction performance and also less network complexity. We discuss two alternatives of concurrency: parallel implementation of the prediction procedure over the ScaLAPACK suite, and the formulation of another parallel routine customized to a higher degree for better performance in the case of the QR-cp procedure.

1 Introduction

Our main objective is to show the application of parallel computing platforms for concurrent execution of an improved prediction algorithm that we have devised. The algorithm is based on RBF (Radial Basis Function) artificial neural network theory, and also on matrix decompositions drawn from linear algebra theory [GL96,Ste01]. Neural network complexity, as described in previous references dealing with this hybrid method [SOPP01b], is enhanced, yielding quite good prediction results that can be now accessible by low-budget research companies or institutes, thanks to the concepts and tools discussed in this paper.

First, the use of the matrix decomposition schemes used is described briefly in order to propose a new means of creating optimal RBF neural architectures for time series prediction applications, one that solves most of the problems observed in the previously described algorithms. Then we will discuss feasible parallelization methods and ideas about their efficiency and implementation.

2 Matrix approach to time series prediction problems

The work in this paper is derived from the consideration of the problem of *time series prediction*, in which there is required an estimate (or *prediction*) of the future value $x(t + t_h)$ of a variable x, where $t_h > 0$ refers to the 'time horizon' (that is, the time period covered by the prediction). In this framework, *endogenous* prediction is computed from the actual and past values $x(t), x(t -$

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 425-432, 2003.

2), ... up to a 'window size' W (that is, up to $x(t-W+1)$) available at time instant t, giving an estimate $\hat{x}(t+t_h)$ of the future value the process or series is assumed to take. In the algorithm we propose, we set up data matrices of size $2W \times W$ in each time instant t. To do so, we use lagged values using consecutive past windows as follows :

$$\mathbf{A}_W\big(x(\cdot),t\big) = \begin{pmatrix} x(t-3W+2) & \cdots & x(t-2W+1) \\ x(t-3W+3) & \cdots & x(t-2W+2) \\ \vdots & \vdots & \vdots \\ x(t-W+1) & \cdots & x(t) \end{pmatrix} , \tag{1}$$

so that we would need, in this example, $3W-1$ past values to predict the value in $t+t_h$. The main idea is using computation on these matrices to determine the relevant "lags" (time indexes) for prediction. This allows us to automatically identify the number of inputs of, say, a neural network model.

3 RBF predictor with matrix decompositions

In the particular context of the RBF networks, the mapping of a time series prediction problem to the network is performed setting up the input as past values (consecutive ones, in a first approximation) of the time series. The output is viewed as a prediction for the future value that we want to estimate, and the computed error between the desired and network- estimated value is used to adjust parameters in the network.

Using ideas from linear algebra theory [SB73], we have derived an algorithm based on RBF networks that is very well suited for solving time series prediction problems. This algorithm, thoroughly described in the specialized reference [SOPP01b], determines the 'optimum' number (in practical terms) of RBF neurons for computing the real-time recursive prediction of complicated time series such as the Mackey-Glass chaotic series [MG77]. In the experiment described in the cited reference, the result (depicted in Figure 2) indicates a substantial reduction to less than half of the neural resources originally required to 'learn' the prediction surface on an on-line basis, when our method based on SVD (and QR-cp, another decomposition coupled with SVD for the experiments, which can also be employed for model reduction) was tested.

We are mainly concerned with the parallelization of a definite subpart of the algorithm, so we will not need to enter into detail concerning its entire operation. Thus we limit ourselves to a brief outline of the underlying ideas and some of the results obtained; the interested reader may refer to [SOPP01b] for a detailed discussion and additional mathematical derivations. Document [SOPP01a] contains some more extensive proofs that mathematically support the use of the SVD and QR-cp techniques for optimal neural network construction.

The basic procedure involves forming a matrix $\mathbf{A}$ in which successive rows set up from neural activation $a_i(\cdot)$ values as real-time input patterns $\mathbf{x}$ are presented to the network. As the output layer is a linear expansion $\sum h_i \cdot a_i(\mathbf{x})$ of these

activations, SVD enables the determination of the most relevant nodes (RBFs) in the network, just by examining the singular values of the related matrix. This is possible since the QR-cp phase transforms relevant singular values σ_i into a permutation matrix $\mathbf{P}$ that indicates the neurons (columns) responsible for most of the data 'energy' [SOPP01a]. This procedure corresponds to the "pruning" subtask depicted in Figure 1. As it has been suggested before, the SVD and QR-cp matrix routines are also used, within the proposed prediction algorithm, to determine the lag structure for prediction (second stage in Figure 1).

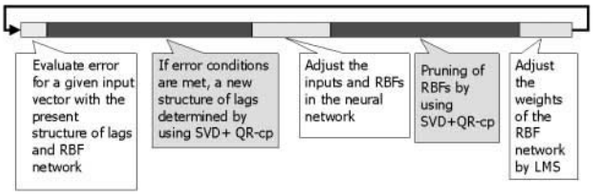


Figure 1. These are the succesive tasks to be sequentially executed in the prediction algorithm that we propose. Steps second and fourth (in darker boxes) amount for the main computational fraction of the whole iteration. As these two steps are based on SVD and QR-cp matrix decompositions, any way for their parallel execution allows to execute the whole algorithm much faster.

We set up activation matrices for the "pruning" routine every few iterations; this allows for the fast detection of dynamics changes in the input time series [SOPP01b]. It can be seen that a lot of matrix work is required for real-time operation of our algorithm. The prediction accuracy obtained must be balanced against this amount of computation. Parallelization of the SVD and QR-cp subtasks offers a way of doing this without too great a sacrifice of the prediction precision obtained.

4 Ideas for Parallelization

We could parallelize matrix routines in two ways: either by using publicly available routines such as the ScaLAPACK suite (to be introduced further on), or by devising a new customized routine for the QR-cp step, because this latter option can be used to better optimize the load distribution of computation evenly between the worker processors, by exploiting the Householder transformations that form the core of the QR-cp step [GL96]. In either case, it should be noted that the study of the parallelization of the SVD and/or QR-cp routines can be done independently of the remaining discussion (and details) concerning our neural prediction algorithm [SOPP01b] because these matrix operations are invoked in the neural *pruning* (node selection) routine, simply by passing successive matrix

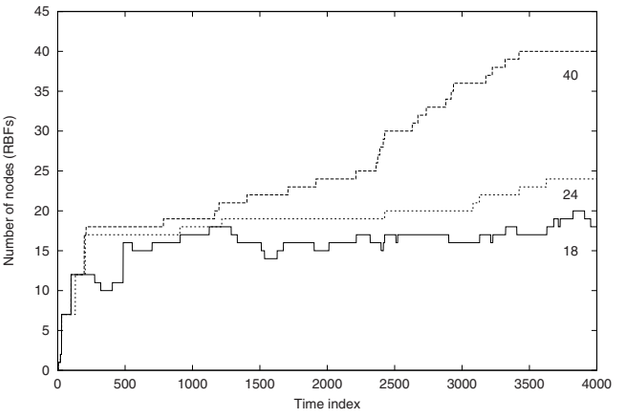


Figure 2. Comparison of the number of neurons (RBFs) required for on-line prediction of the chaotic Mackey-Glass series (upper plot = usual (RAN) algorithm; lower plots = two versions of the algorithm that use SVD and QR-cp techniques for model reduction).

operands to them, and receiving the same kind of return information. Parallelization of SVD is widely implemented, but in what regards QR-cp, we need a different approach because of the data distribution issue, with we are showing as more efficient when we consider a new method based on row distribution.

It is advisable to resort to a 'customized' version of the QR-cp factorization, since most of the parallel implementations available refer to the general (not column pivoted) QR decomposition. Besides, the QR-cp returns the permutation factor $\mathbf{P}$ [GL96] required for correct operation of our RBF neuron selection algorithm [SOPP01b]. In this sense, it is important to take into account the *order* in which the columns must be processed in QR-cp, which is not necessarily the natural (1 to n) order. This is why a column-wise distribution among the processors is not efficient because, even if it were performed in a homogeneous way, the volume of communications might be too great. Thus, in our parallel scheme we have started from a different point of view, one as far as we know that is not explicitly described elsewhere, for treating QR-cp parallelization. Our scheme is based on a *row-wise* distribution of the argument matrix. The advantages of this approach are manifold. First, local computation in each of the remaining processors is performed over a subset of rows of the original matrix, taking into account the successive columns to be affected. Even if not processed in the natural order, the order established would be the same for all processors. Our procedure a llows a more equitative partitioning of the load among the distinct processing nodes, because all of them would be working over rows of equal length (n). This differs from the classic column-wise distribution in which, depending on the pattern of the QR-cp loop over the concrete input matrix, some processors could get 'unfair' treatment with respect to others, as regards local computation.

The general QR factorization of a matrix $\mathbf{A}$ can be computed by applying a sequence of 'zeroing' orthogonal Householder transformations, that can be comprised in a single matrix

$$\begin{aligned}\mathbf{Q} &\equiv [\mathbf{H}_k \cdots \mathbf{H}_2 \mathbf{H}_1]^T \\ &= \mathbf{H}_1 \mathbf{H}_2 \cdots \mathbf{H}_k \ , \end{aligned} \tag{2}$$

so that $\mathbf{Q}^T\mathbf{A} = \mathbf{R}$, that is, $\mathbf{A} = \mathbf{QR}$, taking into account that the product of every pair of orthogonal matrices is again an orthogonal matrix. If we use the notation $\mathbf{A}_{k+1} \equiv \mathbf{H}_k\mathbf{A}_k$, where $k = 1, 2, \ldots, n$, then[1] the Householder vectors have to be chosen so that when $\mathbf{Q}_k$ premultiplies matrix $\mathbf{A}_k$, it sets to zero all the elements $\mathbf{A}_k(i,k)$ with $i > k$ (the $m - k$ elements in column k, below the main diagonal). And so the result of the application of $\mathbf{Q}$ will be an upper-triangular matrix, as required in QR [Bjö96]. In the case of QR-cp, the algorithm must apply n Householder transformations $\mathbf{H}_k$ $(k = 1, 2, \ldots, n)$ and n permutation matrices $\mathbf{P}_k$. Each of the latter must perform a single interchange with respect to the identity $\mathbf{I}_n$, between columns p and k, where

$$p = \min \left\{ i \in \mathbb{N}, k \leq i \leq n : \|\mathbf{r}_i\| = \max_{k \leq s \leq n} \{\|\mathbf{r}_s\|\} \right\} \ , \tag{3}$$

with

$$\mathbf{A}_k = \begin{pmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{R}_{22} \end{pmatrix} \tag{4}$$

with the transformed matrix being up to the k-th step, that is,

$$\mathbf{A}_k = \mathbf{H}_{k-1} \cdots \mathbf{H}_2 \mathbf{H}_1 \mathbf{A} \mathbf{P}_1 \mathbf{P}_2 \mathbf{P}_{k-1} \tag{5}$$

and $\mathbf{R}_{22} = [\mathbf{r}_k \mathbf{r}_{k+1} \cdots \mathbf{r}_n]$. With this background, it can be shown [GL96] (Section 5.4.1) that the algorithm whose pseudocode is described in Figures 3 and 4 performs the QR-cp factorization of $\mathbf{A}$.

When the algorithm ends, the matrix $\mathbf{A}$ is replaced by the upper-triangular matrix $\mathbf{R}$ from the QR-cp factorization of $\mathbf{A}$. We stress that matrix $\mathbf{Q}$ is not explicitly or implicitly needed in our algorithm [SOPP01b], because it is based only on the permutation matrix $\mathbf{P}$ that represents the successive indexes from the column pivotings.

5 Experimental results

Experimental results of execution of the proposed algorithm over a 'cluster' of personal computers are presented next. The kind of parallel architecture considered here is a small cluster of personal computers. The hardware configuration of the cluster is as follows: there is one console node (the *host*) with the common file system. The 'master' portion of the algorithm was executed in this node, based on a Pentium II CPU running at 400 MHz with 128 MB main memory. Eight (8)

[1] We must set $\mathbf{A}_1 \equiv \mathbf{A}$, and then $\mathbf{A}_{n+1} \equiv \mathbf{R}$ holds.

Master Processor:

1. Set $r = 1$, and $\mathbf{P} = \mathbf{I}_n$.
2. Compute the square of the 2-norm of the r-th to n-th columns from matrix $\mathbf{A}$. Denote those values by $\{c_s\}$ $(r \leq s \leq n)$.
3. Determine the lowest index k such that $c(k) = \max_{r \leq i \leq n} c(i)$.
4. Interchange columns r and k of $\mathbf{A}$, and columns r and k of $\mathbf{P}$.
5. Compute the normalized Householder vector $\mathbf{v}$ (that is, with first component equal to 1) and the β parameter that sets to zero the set of elements $\mathbf{A}(i, r)$, for $r + 1 \leq i \leq m$. An algorithm to determine β and the Householder vector required $\mathbf{v}$ is presented, e.g., in Section 5.1.3 of [GL96]. This algorithm, moreover, rewrites the diagonal element $\mathbf{A}(r, r)$ with the 2-norm of the original subcolumn that consisted of the elements $\mathbf{A}(i, r)$ $(r \leq i \leq m)$.
6. Send β and $\mathbf{v}$ to all the slave processors ('broadcast'), as well as the submatrix formed by $\mathbf{A}(i, j)$, where $r \leq i \leq m$, $r \leq j \leq n$ (the entire submatrix is sent to every processor).
7. Receive the updated submatrix (only the rows assigned to each processor are received from that processor). Note that in our case we do not need to preserve the Householder vectors $\mathbf{v}$ as is done in [GL96] (Section 5.4.1). These can be regarded as a 'compact' representation of $\mathbf{Q}$, while we are only interested in the resulting permutation $\mathbf{P}$, which is locally computed in this master processor.
8. $r \longleftarrow r + 1$. If $r > n$ then stop; the desired permutation is contained in $\mathbf{P}$.
9. Determine the new values for the local parameters $\{c_s\}$, according to the formula

$$c(s) \longleftarrow c(s) - \sum_{j=r}^{n} \left[\mathbf{A}(r-1, j)\right]^2 , \quad r \leq s \leq n \ . \tag{6}$$

10. Go back to step (2).

Figure 3. Pseudocode description of the master process operation.

Slave processors:

1. Receive β, the Householder vector $\mathbf{v}$, and the submatrix $\mathbf{A}$.
2. Update *rows corresponding to this processor* (a homogeneous distribution of rows among the processors is assumed) according to

$$\mathbf{A}(i, j) \longleftarrow \mathbf{A}(i, j) + \beta \cdot \mathbf{v}(i) \cdot \sum_{s=1}^{m} \mathbf{A}(s, j) \cdot \mathbf{v}(s) \ , \tag{7}$$

   where m and n are the dimensions of the received matrix $\mathbf{A}$ Variable i takes the indexes of rows assigned to this local processor, and $1 \leq j \leq n$.
3. Send the locally updated rows back to the master.
4. If $m = n = 1$ then stop; otherwise go back to step (1).

Figure 4. Pseudocode description of the slave process operation.

additional nodes, Pentium II 333 MHz, were used as the 'slaves', and the nine nodes were interconnected through a Fast Ethernet (100 Mbits per sec.) switch. In order to perform the distribution and sends/receives of the data and submatrices to every processor, we made use of the *Distributed and Parallel Application Toolbox* (DP-Toolbox) publicly available for the MATLAB environment[2]. Besides, this complement to MATLAB allows the execution of a local copy of the MATLAB environment in each slave, and so low level operations were performed in the MATLAB language within the slaves, but again using a loop unfolding to isolate the optimizing effect of the MATLAB environment. Runs were made for the QR-cp decomposition for matrix sizes of 40×40, 100×100, and 200×200 (5 runs for each size, then calculating the sample mean). The data used in the matrices (the source of which, as we know, is not relevant for the results obtained) were from neural activations for the Mackey-Glass time series prediction drawn from the execution of the prediction algorithm mentioned in this paper. As stated previously, the sample mean T_P was computed from the execution time for every 5 runs using a given number $T_P \in \{2, 4, 6, 8\}$ of slaves and every size example considered. The speedup was computed as the quotient $S_P = T_S/T_P$, where T_S denotes the sample mean for 5 executions of the sequential version, for the three sizes considered. Efficiency of use of the parallel system (the ratio S_P/P) was also computed for each experiment. The results are summarized in Table 1. In the $N_P = 8$ case, the whole cluster is active.

Table 1. Measures for the speedup and corresponding efficiency for several settings of the QR-cp parallel program (full-programmed version of the described algorithm).

Matrix size	N_P	Speedup	Effic.
40×20	2	1.51	0.755
40×20	4	2.54	0.635
40×20	6	3.15	0.525
40×20	8	3.48	0.435
100×50	2	1.59	0.795
100×50	4	3.10	0.775
100×50	6	4.51	0.752
100×50	8	5.68	0.710
200×100	2	1.81	0.905
200×100	4	3.56	0.890
200×100	6	5.26	0.877
200×100	8	6.78	0.848

As expected, the algorithm performs better with larger matrix sizes, as this implies a higher granularity in the local computations. The net effect is a minimization of the impact of communication latency and its impact on the measures.

[2] Internet: `http://www-at.e-technik.uni-rostock.de/dp/`.

It also produces a better exploitation of the bandwidth offered by the switch, because of the formation of messages with higher informative content, with respect to the overheads from headers and other elements of the communication protocol. The slight performance degradation when the number of slave processors rises (due to the communication overhead) becomes smoother as the problem size grows.

6 Conclusions

Clusters of computers have become a practical choice for improving the performance of engineering and mathematically oriented applications in industrial and commercial environments requiring inexpensive platforms. We have shown the efficent application of parallel computation to the time series prediction problems that arise in many applications of economic and social interest, such as the management of natural resources, optimal planning and financial tools. Our prediction algorithm, described in full detail in previous literature, relies on neural networks (RBF networks), but as a quite high amount of realtime data is to be processed, matrix operations should be parallelized in order to obtain the good and accurate results reported. Our discussion shows that there is an interplay between mathematical, computational, and artificial intelligence tools that cooperate together to solve (with low-cost) practical prediction problems.

7 Acknowledgements

This work was partially supported by Spanish CICYT Projects TIC2000-1348 and TIC2001-2845.

References

[Bjö96] Å. Björck. "Numerical Methods for Least Squares Problems". SIAM Publications. Philadelphia, U.S.A. (1996).

[GL96] G.H. Golub and C.F. Van Loan. "Matrix Computations". The Johns Hopkins University Press. Baltimore, Maryland, U.S.A., 3rd edition (1996).

[MG77] M.C. Mackey and L. Glass. Oscillation and chaos in physiological control systems. *Science* **197**, 287–289 (1977).

[SB73] H. Schneider and G.P. Barker. "Matrices and Linear Algebra". Dover Publications. New York, U.S.A. (1973).

[SOPP01a] M. Salmerón, J. Ortega, C.G. Puntonet, and F.J. Pelayo. "Time Series Prediction with Hybrid Neuronal, Statistical and Matrix Methods (in Spanish)". Department of Computer Architecture and Computer Technology. University of Granada, Spain (2001).

[SOPP01b] M. Salmerón, J. Ortega, C.G. Puntonet, and A. Prieto. Improved RAN sequential prediction using orthogonal techniques. *Neurocomputing* **41**(1–4), 153–172 (2001).

[Ste01] G.W. Stewart. "Matrix Algorithms. Volume II: Eigensystems". SIAM Publications. Philadelphia, U.S.A. (2001).

New Method for Filtered ICA Signals Applied To Volatile Time Series.

J.M. Górriz, Carlos G. Puntonet, Moisés Salmerón, and Julio Ortega

Dpto. Ingeniería de Sistemas y Automática,
Tec. Electrónica y Electrónica. Universidad de Cádiz, and
Dpto. Arquitectura y Tecnología de Computadores,
Universidad de Granada
E-mail: juanmanuel.gorriz@uca.es

Abstract. In this paper we propose a new method for volatile time series forecasting using techniques like Independent Component Analysis (ICA) or Savitzky-Golay filtering as preprocessing tools. The preprocessed data will be introduce in a based radial basis functions (RBF) Artificial Neural Network (ANN) and the prediction result will be compared with the one we get without these preprocessing tools or the classical Principal Component Analysis (PCA) tool.

1 Introduction

Different techniques have been developed in order to forecast time series using data from the stock. There also exist numerous forecasting applications like those ones analyzed in [17]: signal statistical preprocessing and communications, industrial control processing, Econometrics, Meteorology, Physics, Biology, Medicine, Oceanography, Seismology, Astronomy y Psychology.

A possible solution to this problem was described by Box and Jenkins [7]. They developed a time-series forecasting analysis technique based in linear systems. Basically the procedure consisted in suppressing the non-seasonality of the series, parameters analysis, which measure time-series data correlation, and model selection which best fitted the data collected (some specific order ARIMA model). But in real systems non-linear and stochastic phenomena crop up, thus the series dynamics cannot be described exactly using those classical models. ANNs have improved results in forecasting, detecting the non-linear nature of the data. ANNs based in RBFs allow a better forecasting adjustment; they implement local approximations to non-linear functions, minimizing the mean square error to achieve the adjustment of neural parameters. Plattś algorithm [16], RAN (Resource Allocating Network), consisted in the control of the neural network's size, reducing the computational cost associated to the calculus of the optimum weights in perceptrons networks. Matrix decomposition techniques have been used as an improvement of Platt model [24] with the aim of taking the most relevant data in the input space, for the sake of avoiding the processing of non-relevant information (NAPA-PRED "Neural model with Automatic Parameter Adjustment for PREDiction"). NAPA-PRED also includes neural pruning [23].

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 433-440, 2003.

The next step was to include the exogenous information to these models. There are some choices in order to do that; we can use the forecasting model used in [9] which gives good results but with computational time and complexity cost; Principal Component Analysis (PCA) is a well-established tool in Finance. It was already proved [24] that prediction results can be improved using the PCA technique. This method linear transform the observed signal into principal components which are uncorrelated (features), giving projections of the data in the direction of the maximum variance [15]. PCA algorithms use only second order statistical information; Finally, in [3] we can discover interesting structure in finance using the new signal-processing tool Independent Component Analysis (ICA). ICA finds statistically independent components using higher order statistical information for blind source separation ([4], [13]). This new technique may use Entropy (Bell and Sejnowski 1995, [6]), Contrast functions based on Information Theory (Comon 1994, [8]), Mutual Information (Amari, Cichocki y Yang 1996, [2]) or geometric considerations in data distribution spaces (Carlos G. Puntonet 1994 [19],[26], [1], [20], [21]), etc. Forecasting and analyzing financial time series using ICA can contributes to a better understanding and prediction of financial markets ([22],[5]).

2 Basic ICA

ICA has been used as a solution of the blind source separation problem of a linear mixture [12] denoting the process of taking a set of measured signal in a vector, $\mathbf{x}$, and extracting from them a new set of statistically independent components (ICs) in a vector $\tilde{\mathbf{s}}$. In the basic ICA each component of the vector $\mathbf{x}$ is a linear instantaneous mixture of independent source signals in a vector $\mathbf{s}$ with some unknown deterministic mixing coefficients:

$$x_i = \sum_{i=1}^{N} a_{ij} s_j \tag{1}$$

Due to the nature of the mixing model we are able to estimate the original sources $\tilde{s}_i$ and the de-mixing weights b_{ij} applying i.e. ICA algorithms based on higher order statistics like cumulants.

$$\tilde{s}_i = \sum_{i=1}^{N} b_{ij} x_j \tag{2}$$

Using vector-matrix notation and defining a time series vector $\mathbf{x} = (x_1, \ldots, x_n)^T$, $\mathbf{s}$, $\tilde{\mathbf{s}}$ and the matrix $\mathbf{A} = \{a_{ij}\}$ and $\mathbf{B} = \{b_{ij}\}$ we can write the overall process as:

$$\tilde{\mathbf{s}} = \mathbf{Bx} = \mathbf{BAs} = \mathbf{Gs} \tag{3}$$

where we define $\mathbf{G}$ as the overall transfer matrix. The estimated original sources will be, under some conditions included in Darmois-Skitovich theorem, a permuted and scaled version of the original ones. Thus, in general, it is only possible

to find $\mathbf{G}$ such that $\mathbf{G} = \mathbf{PD}$ where $\mathbf{P}$ is a permutation matrix and $\mathbf{D}$ is a diagonal scaling matrix.

In Financial time series this model (equation 1) can be applied to the stock series where there are some underlying factors like seasonal variations or economic events that affect the stock time series simultaneously and can be assumed to be quite independent [14].

3 Preprocessing Volatile Signals

The main goal, in the preprocessing step, is to find non-volatile time series including exogenous information i.e. financial time series, easier to predict using ANNs based on RBFs. This is due to smoothed nature of the kernel functions used in regression over multidimensional domains [11]. We propose the following Preprocessing Steps

- After Whitening the set of time series $\{x_i\}_{i=1}^{n}$(subtract the mean of each time series and removing the second order statistic effect or covariance matrix diagonalization process)
- we apply some ICA algorithm to estimate the original sources s_i and the mixing matrix $\mathbf{A}$ in equation 1. Each IC has information of the stock set weighted by the components of the mixing matrix. In particular, we use an equivariant robust ICA algorithm based in cumulants. The de-mixing matrix is calculated according the following iteration:

$$\mathbf{B}^{(n+1)} = \mathbf{B}^{(n)} + \mu^{(n)}(\mathbf{C}_{\mathbf{s},\mathbf{s}}^{1,\beta}\mathbf{S}_{\mathbf{s}}^{\beta} - I)\mathbf{B}^{(n)} \tag{4}$$

  where $\mathbf{I}$ is the identity matrix, $\mathbf{C}_{\mathbf{s},\mathbf{s}}^{1,\beta}$ is the $\beta+1$ order cumulant of the sources (we chose $\beta = 3$ in simulations) and $\mathbf{S}_{\mathbf{s}}^{\beta} = diag(sign(diag(\mathbf{C}_{\mathbf{s},\mathbf{s}}^{1,\beta})))$. Once convergence, which is related with cross-cumulants [1] absolute value, is reached , we estimate the mixing matrix inverting $\mathbf{B}$.
- Filtering.
  1. We neglect non-relevant components in the mixing matrix $\mathbf{A}$ according to their absolute value. We consider the rows $\mathbf{A}_i$ in matrix $\mathbf{A}$ as vectors and calculate the mean Frobenius Norm [2] of each one. Only the components bigger than mean Frobenius Norm will be considered. This is the principal preprocessing step using PCA tool, in this case is not enough.

$$\tilde{\mathbf{A}} = \mathbf{Z} \cdot \mathbf{A} \tag{5}$$

     where $\{\mathbf{Z}\}_{ij} = [\{\mathbf{A}\}_{ij} > \frac{||\mathbf{A}_i||_{Fr}}{n}]$
  2. We apply a low band pass filter to the ICs. We choose the well-adapted for data smoothing Savitsky-Golay smoothing filter [25] for two reasons: *a*)ours is a real-time application for which we must process a continuous

[1] cumulants between different sources

[2] Frobenius Norm: $||\mathbf{x}||_{Fr} \equiv \sqrt{\sum_{i=1}^{n} x_i^2}$

data stream and wish to output filtered values at the same rate we receive raw data and *b*) the quantity of data to be processed is so large that we just can afford only a very small number of floating operations on each data point thus computational cost in frequency domain for high dimensional data is avoided even the modest-sized FFT. This filter is also call Least-Squares [10], and derives from a particular formulation of the data smoothing problem in the time domain and their goal is to find filter coefficients c_n in the expression:

$$\bar{s}_i = \sum_{n=-n_L}^{n_R} c_i s_{i+n} \tag{6}$$

where $\{s_{i+n}\}$ represent the values for the ICs in a window of length $n_L + n_R + 1$ centered on i and $\tilde{s}_i$ is the filter output (the smoothed ICs), preserving higher moments [18]. For each point s_i we least-squares fit a m order polynomial for all $n_L + n_R + 1$ points in the moving window and then set $\tilde{s}_i$ to the value of that polynomial at position i. As shown in [18] there are a set of coefficients for which equation 6 accomplishes the process of polynomial least-squares fitting inside a moving window:

$$c_n = \{(\mathbf{M}^T \cdot \mathbf{M})^{-1}(\mathbf{M}^T \cdot \mathbf{e}_n)\}_0 = \sum_{j=0}^{m} \{(\mathbf{M}^T \cdot \mathbf{M})^{-1}\}_{0j} \cdot n^j \tag{7}$$

where $\{\mathbf{M}\}_{ij} = i^j$, $i = -n_L, \ldots, n_R$, $j = 0, \ldots, m$, and $\mathbf{e}_n$ is the unit vector with $-n_L < n < n_R$.

– Reconstructing the original signals using the smoothed ICs and filtered $\tilde{\mathbf{A}}$ matrix we get a less high frequency variance version including exogenous influence of the old ones. We can write using equations 5 and 4.

$$\mathbf{x} = \tilde{\mathbf{A}} \cdot \bar{\mathbf{s}} \tag{8}$$

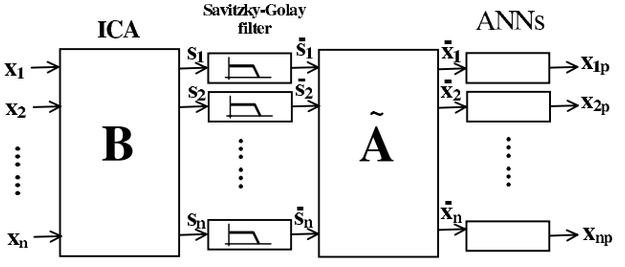


Fig. 1. Schematic representation of prediction and filtering process.

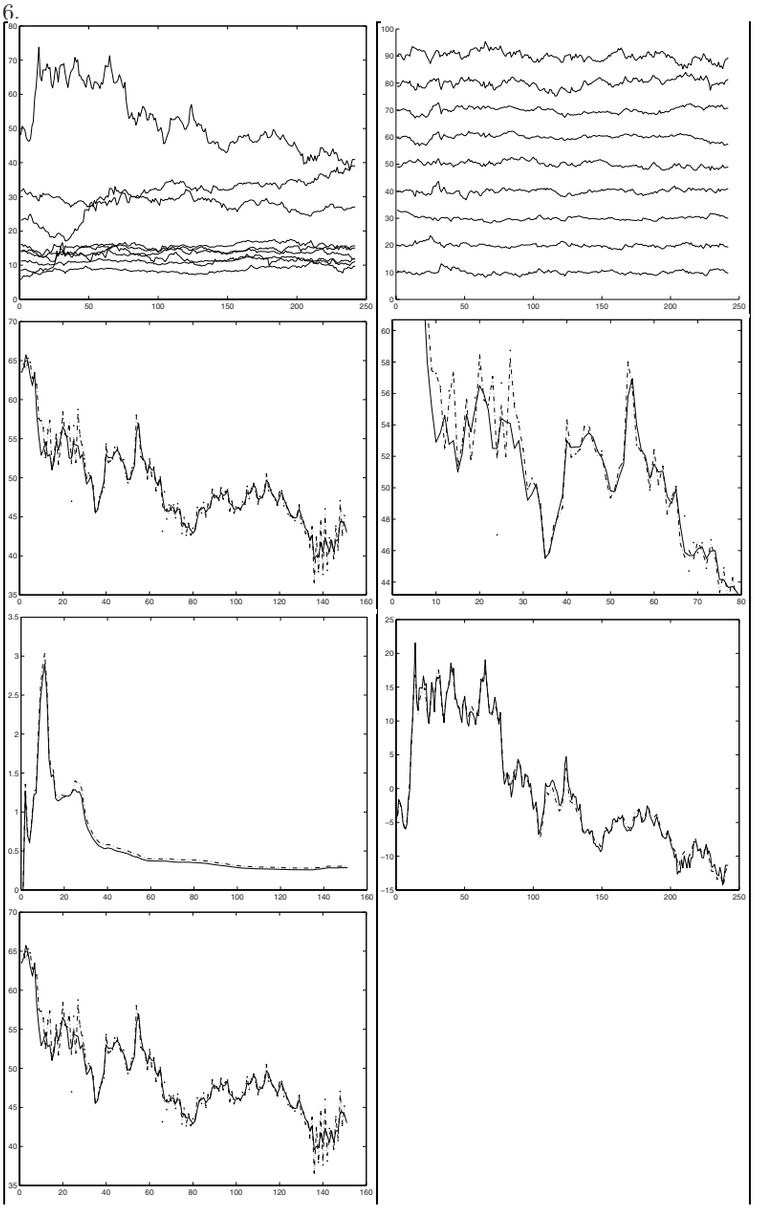
4 Time Series Forecasting Model

We use an ANN using RBFs to forecast a series x_i from the Stock Exchange building a forecasting function $\mathbf{P}$ with the help of NAPA-PRED+LEC algorithm, for one of the set of signals $\{x_1, \ldots, x_n\}$. The individual forecasting function can be expressed in term of RBFs [27] as: $\mathbf{F}(\mathbf{x}) = \sum_{i=1}^{N} f_i(\mathbf{x}) = \sum_{i=1}^{N} h_i exp\{\frac{||\mathbf{x}-\mathbf{c}_i||}{r_i^2}\}$, where $\mathbf{x}$ is a p-dimensional vector input at time $\mathbf{t}$, $\mathbf{N}$ is the number of neurons (RBFs) , $\mathbf{f}_i$ is the output for each neuron i-th , $\mathbf{c}_i$ is the centers of i-th neuron which controls the situation of local space of this cell and $\mathbf{r}_i$ is the radius of the i-th neuron. The global output is a linear combination of the individual output for each neuron with the weight of $\mathbf{h}_i$. Thus we are using a method for moving beyond the linearity where the core idea is to augment/replace the vector input $\mathbf{x}$ with additional variables, which are transformations of $\mathbf{x}$, and then use linear models in this new space of derived input features. RBFs are one of the most popular kernel methods for regression over the domain and consist on fitting a different but simple model at each query point $\mathbf{c}_i$ using those observations close to this target point in order to get a smoothed function. This localization is achieved via a weighting function or kernel $\mathbf{f}_i$. The preprocessing step suggested in section 3 is necessary due to the dynamic of the series [22] and it will be shown that results improve sensitively. Thus we will use as input series the ones we got in equation 8.

5 Simulations and Conclusions.

We work with indexes of different Spanish banks and companies during the same period to investigate the effectiveness of ICA techniques for financial time series (1 in table 2). We have specifically focussed on the IBEX35 from Spanish stock, which we consider the most representative sample of the Spanish stock movements, using closing prices in 2000. We considered the closing prices of Bankinter for prediction and 8 more stock of different Spanish companies (ACS, Aguas de Barcelona, Banco Popular, Banco Santander, BBVA, Dragados, Carrefour and Amadeus). Each time series includes 200 points corresponding to selling days (quoting days). We performed ICA on the Stock returns using the ICA algorithm presented in section 3 assuming that the number of stocks equals the number of sources supplied to the mixing model. This algorithm whiten the raw data as the first step. The ICs are shown in 2 table 2. These ICs represents independent and different underlying factors like seasonal variations or economic events that affect the stock time series simultaneously. Via the rows of $\mathbf{A}$ we can reconstruct the original signals with the help of these ICs i.e. by Frobenius filtering the original mixing matrix is transformed, and we neglect the influence of the corrresponding ICs on the original stock. Thus only a few ICs contribute to most of the movements in the stock returns and each IC contributes to a level change depending its amplitude transient [3]. We do a polynomial fit in the ICs using the library supported by *MatLab* and the reconstruct the selected stock (6 and 7 in table 2) to supply the ANN.

Fig. 2. Simulations.1 Set of Stock Series.2 ICs for the stock series. 3 Real Series(line) Predicted Series with ICA+SG(-.) Predicted Series without preprocessing (.). 4 Zoom 3. 5 NRMSE evolution for ANN with ICA+SG (line) and without (-.). 6 Real Series from ICA reconstruction (scaled old version)(line) and Preprocessed Real Series (-.). 7 Zoom 6.



In 3,4 and 5 in table 2 we show the results we got using our ANN with the above mentioned algorithm. We can say that prediction is better with the preprocessing step avoiding the disturbing peaks or convergence problems in prediction. As is shown in 5 the NRMSE is always lower using the techniques we discussed in section 4.

5.1 Conclusions.

In this paper we showed that prediction results can be improved with the help of techniques like ICA. ICA decompose a set of 9 returns from the stock into independent components which fall in two categories: *a*) large components responsible of the major changes in level prices and *b*) small fluctuations. Smoothing this components and neglecting the non-relevant ones we can reconstruct a new version of the Stock easier to predict. Moreover we describe a new filtering method to volatile time series that are supplied to ANNs in real-time applications.

6 Acknowledegments

This research was partially supported by Spanish CICYT Projects TIC2000-1348 and TIC2001-2845.

References

1. C.G. Puntonet A. Mansour, N. Ohnishi, *Blind multiuser separation of instantaneous mixture algorithm based on geometrical concepts*, Signal Processing **82** (2002), 1155–1175.
2. S. Amari, A. Cichocki, and H. Yang, *A new learning algorithm for blind source separation*, Advances in Neural Information Processing Systems. MIT Press **8** (1996), 757–763.
3. A. D. Back and A. S. Weigend, *Discovering structure in finance using independent component analysis*, Computational Finance (1997).
4. Andrew D. Back and Thomas P. Trappenberg, *Selecting inputs for modelling using normalized higher order statistics and independent component analysis*, IEEE Transactions on Neural Networks **12** (2001).
5. Andrew D. Back and A.S. Weigend, *Discovering structure in finance using independent component analysis*, 5th Computational Finance 1997 (1997).
6. A.J. Bell and T.J. Sejnowski, *An information-maximization approach to blind separation and blind deconvolution*, Neural Computation **7** (1995), 1129–1159.
7. G.E.P. Box, G.M. Jenkins, and G.C. Reinsel, *Time series analysis. forecasting and control*, Prentice Hall, 1994.
8. P. Comon, *Independent component analysis: A new concept?*, Signal Processing **36** (1994), 287–314.
9. J.M. Górriz, J.J.G. dela Rosa, Carlos G. Puntonet, and M. Salmerón Campos, *New model for time-series forecasting using rbfś and exogenous data*, In Press (2003).
10. R.W. Hamming, *Digital filters*, 2ª ed., Prentice Hall, 1983.
11. T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning*, Springer, 2000.

12. A. Hyvärinen, J. Karhunen, and E. Oja, *Independent component analysis*, John Wiley and Sons, 2001.
13. A. Hyvärinen and E. Oja, *Independent component analysis: Algorithms and applications*, Neural Networks **13** (2000), 411–430.
14. K. Kiviluoto and E. Oja, *Independent component analysis for paralell financial time series*, Proc. in ICONIP98 **1** (1998), 895–898.
15. T. Masters, *Neural, novel and hybrid algorithms for time series analysis prediction*, John Miley & Sons, 1995.
16. J. Platt, *A resource-allocating network for function interpolation*, Neural Computation **3** (1991), 213–225.
17. D.S.G. Pollock, *A handbook of time series analysis, signal processing and dynamics*, Academic Press, 1999.
18. W.H. Press, S.A. Teukolsky, W.T. Vertterling, and B.P. Flannery, *Numerical recipes in c++*, 2^{a} ed., Cambridge University Press, 2002.
19. C.G. Puntonet, *Nuevos Algoritmos de Separación de Fuentes en Medios Lineales*, Ph.D. thesis, University of Granada , Departamento de Electrónica y Tecnología de Computadores, 1994.
20. C.G. Puntonet and Ali Mansour, *Blind separation of sources using density estimation and simulated annealing*, IEICE Transactions on Fundamental of Electronics Communications and Computer Sciences **E84-A** (2001).
21. M. Rodríguez-Álvarez, C.G. Puntonet, and I. Rojas, *Separation of sources based on the partitioning of the space of observations*, Lecture Notes in Computer Science **2085** (2001), 762–769.
22. J.M. Górriz Sáez, *Predicción y Técnicas de Separación de Senales*, Ph.D. thesis, University of Cádiz , Departamento de Ing. de Sistemas y Aut. Tec. Eleectrónica y Electrónica, 2003.
23. Moisés Salmerón, Julio Ortega, Carlos G. Puntonet, and Alberto Prieto, *Improved ran sequential prediction using orthogonal techniques*, Neurocomputing **41** (2001), 153–172.
24. M. Salmerón-Campos, *Predicción de Series Temporales con Redes Neuronales de Funciones Radiales y Técnicas de Descomposición Matricial*, Ph.D. thesis, University of Granada , Departamento de Arquitectura y Tecnología de Computadores, 2001.
25. A. Savitzky and M.J.E. Golay, Analytical Chemestry **36** (1964), 1627–1639.
26. F. J Theis, A. Jung, E.W. Lang, and C.G. Puntonet, *Multiple recovery subspace projection algorithm employed in geometric ica*, In press on Neural Computation (2001).
27. J. Moody y C. J. Darken, *Fast learning in networks of locally-tuned processing units*, Neural Computation **1** (1989), 284–294.

R b f de ce e e e ed e e e

R l R m n ll n n l

ni i a cnica ic an a a ia In a ica
a i a 0 a a ai i
{ l n h ll n n u l
ni i y un a n u ci nc
un any
un un

s c i cia n u a n a n i y u in
i n ic i n an i i u ua y i a a
c n nc un i ac n ic a u nu
a n i a in unc ain y a ci
a i a a u ic y a a n u a n ,
u a c u a i na y in n i n y a i un
c ain a u i n , ic a a y a i in ac ic n
ica u a u c n uc i n c n nc in a in
ic i n n nin a i i y , i
i a n i a a a a u a nin a i a
a i a i n n a a i c n a ina c n nc
in a a i c n uc u y In u nc
unc i n i a n a u c ni u n c u
n a i i a a
R a i cia u a , u a n
in a i , n nc In a , i i

o c o

w l l w k b n w l l
m ll n l l l n y m n b bl
m l b l y n m l bl m b n n n
n l n w k b m b l n n n l I
m n lly n n b n b n q y n
w m n m n n l n n 11 w ll n n
n l n w k 10 l bl m n n I n ly
lly b q m 9 1 m n
ny n m n n n m l n l

---

* i a u in a y a c an n cy 0 0 0 an 0 0 0 ,
in a y a c an H 0 an ini y uca i n an
a c an in a y a c an I an in a y
In i an I I I

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 441-448, 2003.

wn n m ly w k l n n l m
m m n n l m l l n n w n
n m n w l y n nfl n b ly b n n n
n m 1 1 n m n
b l m l n n b n b m
m l m 8 m l b l y n
w n R b n n In l by y n m n ly l
m n m l
In n w w m m l n l n m l m

F o a c a a o

n l m n y n l m n ll n n n
n y l y n n n l y w
l l b w n n n l y n mb n
n n m n by l n l nk n n
m n l y w ny y b l l b k
nn n m l y n l n l y w nly n
n n l n l m l n b
by n mb n n n by S { _ $\underline{w}$ $\mathbb{R}$ _ $\mathbb{R}^m$ $\underline{w}$ }
w $\mathbb{R}^d$ m n m n l n
b n _ $\underline{w}$ n n l n n n _ w $\underline{w}$ w w $w_d{}^T$ b n
m n mb n n n n 1
n mb m l n n n b ly
n w n mb l y n n n
G n m l b n $\{\_^t{}_t\}_{t\ ..n}$ w w
n y m $_t$ w n l n n m
bl _ n n $\_^t$ n m m
n nkn wn n n n _ _ ny
l _ m n by _ _ w
n m bl w n n n k n l
l n n n n m _ $\underline{w}$ nkn wn n n _ by

$\hat{}$ _ $\underline{w}$ $\sum w$ $\sum^m w$ w_m , w 1

w $\underline{w}$ m b m $_s$ l n y n n l n y
n n l m n mb n n n m n
n n n l m l n n n
$_s$ n b ny n n l n y l n y n n b n n
n bl n n n n n y lly
q n l b n n n l y q n n n
l n n 1 e { } n n m mm nly
n n n n l b l n n n
q n n n

o a o F

In m l w k wn m l w n
ly n b n n w y l n n n n
b n m y m n In
m m l w n n n l m b n m
$\{\_^{t}\ _{t}\}_{t\ ..n}$ n n n n n n lly
b m l m n m nkn wn b b l y
b n $\_$ n nl n n n $\_$ m m
by $\_\ \underline{w}$ r w $\mathbb{R}$ $\_$ $\mathbb{R}^m$
n $\underline{w}$ $\mathbb{R}^d$ nkn wn m n r $\mathbb{R}$
l n n l n n m bl

$\_$
m $\underline{\hat{w}}_n$ n by $\underline{\hat{w}}_n$ r $\{R_{\ n}\ \underline{w}\ \ \underline{w}\ \}$
$\mathbb{R}^d$ w $R_{\ n}\ \underline{w}$ b n n l n by ll w n q n

$$R_{\ n}\ \underline{w}\quad \frac{1}{\_}\sum_{t}^{n}\ _{t}\quad \_^{t}\ \underline{w}$$

w b n n n b n nfl n
n l n m n n bl w
n by ψ r $\underline{w}$ $\frac{\partial\ r_{\_}}{\partial r}$ m $\underline{\hat{w}}_n$ n b n
m l y by l n ll w n q n

$$\sum_{t}^{n}\psi\ _{t}\quad \__{t}\ \underline{w}\quad \__{t}\ \underline{w}\quad \underline{0}$$

w ψ $\mathbb{R}$ $\mathbb{R}$ r_t $_t$ $\_^{t}\ \underline{\hat{w}}_n$ l n

$$\_\ \underline{w}\quad \overline{w}\quad \_\ \underline{w}\quad \overline{w_d}^{\,T}\quad \_\ \underline{w}$$

n w ll n $\_\ \underline{w}$ In
ll w n w w ll n $\underline{w}$ l b l m m b n by m n m n
n n l R $\underline{w}$

1 I t t t t r t NN

In y b n m w n ly *fl e e*
I m I l l m n by m l
n b n n n m l n m n n n $\_$
n m
I m l m l $\underline{w}$ n l l
b n n n $\_$ n by ll w n q n

$$I\ \_\ r\ \underline{w}\qquad \psi\ r\ \underline{w}\qquad \_\ \underline{w}^{\,T}$$

w r l _ $\underline{w}$ n by q n n n by

$$r \_ \underline{w} \quad \_$$

w n w m $\underline{w}$ $r \_ \underline{w}$
$\psi \; r \; \underline{w}^{T} \; \psi \; r \; \underline{w}$ n $\frac{\partial \; \dot{\_}}{\partial \; \_ \partial \; \_}$ n m
In n b bl n m b m
1 m n n n m

$$\hat{}_n \quad \frac{1}{} \sum_{t}^{n} \quad r_t \; \_t \; \underline{\hat{w}}_n$$

w $\underline{\hat{w}}_n$ m b n m by m n m n
k n n l w n m nfl n n
_ by $\hat{I}$ _ $r \; \underline{\hat{w}}_n \; \psi \; r \; \underline{\hat{w}}_n \; \hat{}_n$ _ $\underline{\hat{w}}_n^{T}$

2 r t t t r t NN

In n n n l n m n
n n by q n 8 1

$$\underline{w} \quad I \_ \underline{w} \quad I \_ \underline{w}^{T} \_ \qquad 8$$
$$\psi \quad Q \; \psi \quad \psi$$

w n by q n n

$$Q \; \psi \quad \psi \; r \; \underline{w} \quad \_ \underline{w} \quad \_ \underline{w}^{T} \_ \qquad 9$$
$$\psi \; r \; \underline{w} \quad \_ \underline{w} \quad \_ \underline{w}^{T}$$

m n n n m Q n n 1 by

$$\hat{Q}_n \quad \frac{1}{} \quad \sum_{t}^{n} \psi \; r_t \; \underline{\hat{w}}_n \quad \_t \; \underline{\hat{w}}_n \quad \_t \; \underline{\hat{w}}_n^{T} \qquad 10$$

l w n m n m $\hat{}$ $\underline{\hat{w}}_n$
$\hat{}_n \; \hat{Q}_n \; \hat{}_n^{T}$ w r _ $\underline{\hat{w}}_n$

o c a

In n w n n n n l n n n l b
n n n w b ym lly l n l
n n w w ll q m $\underline{w}$
bl n n w w n n
n m l n n n _ b n n bl
$\underline{w}$ 1 In ll w n w w ll m k m n m

bl w n l n w k n n n w m y
n n n w n n n l n ll
m w w ll m q n m $\{\underline{\hat{w}}_t\}_{t \ldots T}$ b
n m l n n l m m n m k n n n
w b b l y 1 m $\underline{w}$ q m n
n b n n n 1
m n n by l y n ym m l n
m n l b n n n w
m n n n n ll w n l b n

$$\bar{\ } \ \underline{\hat{w}}_n \quad \underline{w}_{\ n} \quad \underline{0}_d \quad \underline{w} \quad \text{n} \quad \text{b} \quad \text{n} \qquad 11$$

w $\underline{w}$ $Q^{\ T}$ b n m q n 8
w m ym b n n w l
m n by m n
n $\_\ \underline{w}$ n n n n ll $\_$ ll w n
l b n

$$\bar{\ } \quad \_^{t}\ \underline{\hat{w}}_n \quad \_^{t}\ \underline{w}_{\ n} \quad \underline{0}_d \quad \text{n} \quad \text{b} \quad \text{n} \qquad 1$$

q n 1 w ym b n n n
m m l G n b n w m n n
n m

$$\_ \quad \_\ \underline{w}^{\ T}\ \underline{w} \quad \_\ \underline{w} \qquad 1$$

w $\_\ \underline{w}$ n by q n l w b l q
n m l n n n

$$\frac{\_\ \underline{\hat{w}}_n \quad \_\ \underline{w}}{\sqrt{\ \ \_}} \qquad 0\ 1 \qquad 1$$

In n kn wn b n b m by

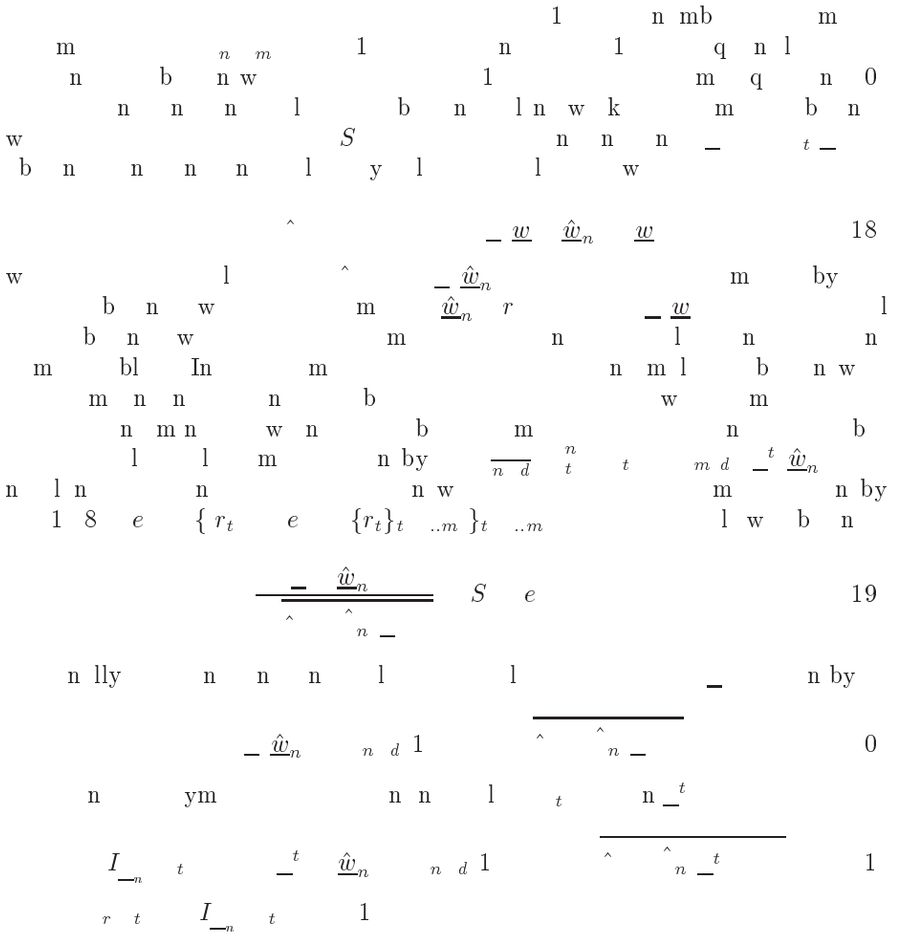
$$\hat{}_{\,n}\ \_ \quad \_\ \underline{\hat{w}}_n^{\ T}\ \hat{} \ \underline{\hat{w}}_n \quad \_\ \underline{\hat{w}}_n \qquad 1$$

l w b n

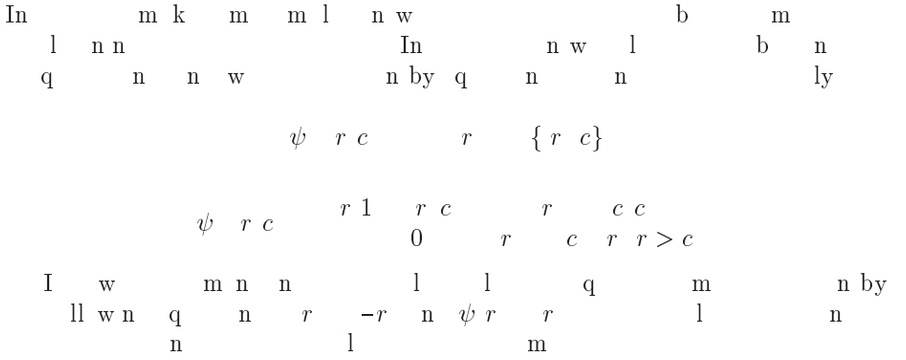
$$\frac{\_\ \underline{\hat{w}}_n \quad \_\ \underline{w}}{\hat{}_{\,n}\ \_} \quad S \quad e \qquad 1$$

100 1 % m n n n l $\_\ \underline{w}$ n by

$$\_\ \underline{\hat{w}}_n \quad {}_{n\ d}\ 1 \qquad \overline{\hat{}_{\,n}\ \_} \qquad 1$$



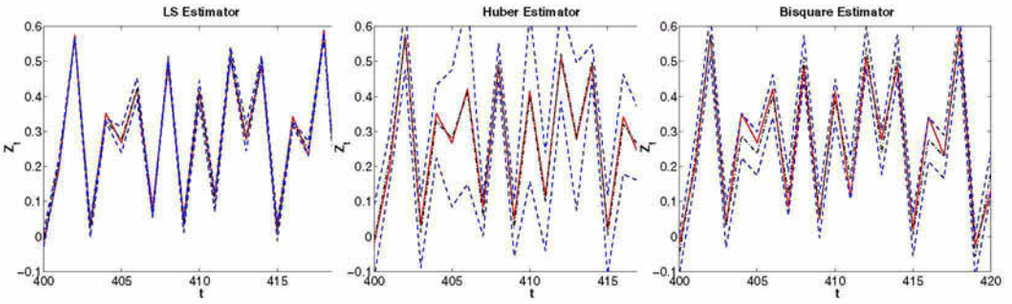
a o



anc an a i i y c a u i u a i
i

s o	o c g s	g s	o c s s	s s
L	0	%	0	0%
H	0 0	0%	0	0 %
B	0	0%	0	0 %

Fig. 1. Confidence Interval of the FANN in the Test set for different estimators. From left to right: Least Square, Huber and Bisquare estimators



References

1. H. Allende, C. Moraga, and R. Salas, *Artificial neural networks in time series forescasting: A comparative analysis*, Kybernetika **38** (2002), no. 6, 685–707.
2. _____, *Robust estimator for the learning process in neural networks applied in time series*, ICANN 2002. LNCS **2415** (2002), 1080–1086.
3. G. Chryssolouris, M. Lee, and A. Ramsey, *Confidence interval prediction for neural network models*, IEEE Transactions of Neural Networks **7** (1996), no. 1, 229–232.
4. J.T. Connor and R.D. Martin, *Recurrent neural networks and robust time series prediction*, IEEE Transactions of Neural Networks **2** (1994), no. 5, 240–253.
5. Richard Golden, *Mathematical methods for neural networks analysis and design*, vol. 1, MIT Press, 1996.
6. F.R. Hampel, E.M. Ronchetti, P.J. Rousseeuw, and W.A. Stahel, *Robust statistics*, Wiley Series in Probability and Mathematical Statistics, 1986.
7. Tom Heskes, *Practical confidence and prediction intervals*, Advances in Neural Information Processing Systems. MIT Press **9** (1997), 176–182.
8. Peter J. Huber, *Robust statistics*, Wiley Series in probability and mathematical statistics, 1981.
9. J. Hwang and A. Ding, *Prediction intervals for artificial neural networks*, J. American Statistical Association **92** (1997), no. 438, 748–757.
10. D. Nix and A. Weigend, *Estimating the mean and the variance of the target probability distribution*, IEEE, in Proceedings of the IJCNN'94 (1994), 55–60.
11. C.S. Qazaz, *Bayesian error bars for regression*, Ph.D. Thesis., Aston University, 1996.
12. I. Rivals and L. Personnaz, *Construction of confidence intervals for neural networks based on least squares estimation*, Neural Networks **13** (2000), no. 1, 463–484.
13. R. Salas, *Robustez en redes neuronales feedforward*, Master's thesis, Universidad Técnica Federico Santa María, 2002.
14. H. J. Sussmann, *Uniqueness of the weights for minimal feedforward nets with a given input-output map*, Neural networks (1992), no. 4, 589–593.
15. Halbert White, *Artificial neural networks: Approximation and learning theory*, Basil Blackwell, Oxford, 1992.

Modelling the HIV-AIDS Cuban Epidemics with Hopfield Neural Networks *

M. Atencia[1], G. Joya[2], and F. Sandoval[2]

[1] Departamento de Matemática Aplicada. E.T.S.I.Informática
[2] Departamento de Tecnología Electrónica. E.T.S.I.Telecomunicación
Universidad de Málaga, Campus de Teatinos, 29071 Málaga, Spain
matencia@ctima.uma.es

Abstract. In this work, Hopfield neural networks are applied to estimation of parameters in a dynamical model of Cuban HIV-AIDS epidemics. The time-varying weights are derived, and its formulation is adapted to the discrete case. The method is tested on a data sequence obtained from numerical solution of the model. Simulation results show that the proposed technique quickly reduces the output prediction error, and it adapts well to parameter changes. Results concerning estimation error are poor, and some directions to deal with this issue are proposed.

1 Introduction

System identification or modelling consists in the determination of the internal properties of a dynamical system, from some measurable external outputs. When no a priori information about the system exists, identification aims at building a model with the only information extracted from observed input-output data. This is an appropriate framework for techniques based upon statistical regression, such as ARMAX and similar methods (see [1] and references therein). On the contrary, when a system model is available, either from physical laws or by our insight into the problem, it is logical not to waste the knowledge provided by the model. In this case, however, the knowledge is not complete, since the numerical value of some parameters of the model must be adjusted. Hence, this task is usually referred to as "gray box" identification of a parametric model [2] or, simply, parameter estimation.

The problem of parameter estimation is often addressed by the gradient method [3], which is an on-line estimator, i.e. it continuously modifies the estimation in the direction of error minimization. However, since the actual value of parameters is unknown, so is the estimation error. Hence, the only measure of the goodness of estimation is the output prediction error, i.e. the difference between the observed output and the predicted output. Consider a general homogeneous dynamical system $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta})$ where $\boldsymbol{\theta}$ is a parameter vector. For

* This work has been partially supported by the Spanish Ministerio de Ciencia y Tecnología (MCYT), Project No. TIC2001-1758. Thanks are due to Hector de Arazoza for providing the model and Liuva Pedroso for generating test data.

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 449-456, 2003.

a particular estimation $\hat{\boldsymbol{\theta}}$, the predicted output is $\boldsymbol{f}(\boldsymbol{x}, \hat{\boldsymbol{\theta}})$ thus the prediction error is given by $\boldsymbol{e} = \dot{\boldsymbol{x}} - \boldsymbol{f}(\boldsymbol{x}, \hat{\boldsymbol{\theta}})$ and the update rule of the gradient method results in $\dot{\hat{\boldsymbol{\theta}}} = -k \nabla \boldsymbol{e}^{\top} \boldsymbol{e}$, where $\boldsymbol{e}$ is regarded as a function of the estimation $\hat{\boldsymbol{\theta}}$. The gain k must be carefully chosen, otherwise slow convergence -too small k- or violent oscillations -too large k- may appear.

The brief presentation of the gradient method in the above paragraph is enough to appreciate that identification can be considered into the wider context of optimization. Thus, optimization techniques are an appealing approach to parameter estimation. In particular, we have applied the optimization methodology of Hopfield and Tank [4], based upon neural networks, to identification of dynamical systems. This *neural estimator* obtains promising results in on-line identification of robotic systems [5], which present several favourable properties: an accurate model stems from physical laws; almost exact measures of states and their derivatives are available from position sensors, speedometers and accelerometer; current sensors provide measures at intervals of nanoseconds, so that the system can be safely considered a continuous model. In this work, we apply this methodology to another problem: modelling the HIV-AIDS epidemics in Cuba. Although the mathematical formulation is similar, the characteristics of this system are completely different: we begin with a model based upon intuition of specialists, rather than physical laws; observed data is obtained from health statistics, which may contain some error; furthermore, the frequency of statistical recounts is, at most, monthly, so that the system is intrinsically discrete. Hence, the estimation technique based upon Hopfield networks faces a formidable challenge when solving this problem, but we think the effort is worthwhile. On one hand, validating an accurate epidemiological model has an obvious importance, since it would allow to test the effectiveness of the decisions of health authorities, as well as to forecast the spread of the disease. In the Cuban case, specifically, it is necessary to determine whether the *contact tracing* program helps in the fight against AIDS. On the other hand, this problem is a good benchmark for estimation with neural networks, as well as for comparison with other techniques. Consequently, this paper aims at determining the main strengths and limitations of neural identification, thus contributing to the choice of an appropriate identification method for each particular problem.

In Sect. 2, the HIV-AIDS model is reproduced and the details of test data generation are explained. Parameter estimation with Hopfield networks is recalled in Sect. 3, and this method is adapted to cope with discrete systems. Section 4 presents the results of simulations and a discussion of the meaning of these results. Finally, concluding remarks are summarized in Sect. 5, together with some lines for future research.

2 A Model of HIV-AIDS Epidemics in Cuba

Infectious diseases have a tremendous impact in human history, and their mathematical modelling has been attempted since centuries, both from the analytical and the stochastic viewpoint (see e.g. [6]). In comparison, the spread of AIDS is a

relatively recent phenomenon, and many facts about this disease are still poorly understood. Hence, the construction of models of HIV-AIDS epidemics would be a valuable tool in order to forecast the infection advance, as well as to assess the policies of health authorities. A crucial characteristic of AIDS with severe epidemiological consequences is its long and variable incubation period: individuals infected by HIV, who remain at a subclinical stage and are not aware of their own infection, can infect other people. Conversely, contagion is less probable when symptoms are apparent, which is the stage properly referred to as AIDS. In an attempt to control proliferation of the infection, Cuban authorities are carrying out the active search of HIV infected individuals. This program encourages those individuals that become symptomatic to identify all their sexual partners, who are analysed in order to find out if they are infected. If this is the case, they likewise declare their sexual contacts, hence the name of the program: *contact tracing*. Also, they receive information on prophylactic measures so as to avoid further spreading the infection.

In order both to forecast the growth of the infection and to assess the contact tracing program, a model of HIV-AIDS has been defined for the specific Cuban case. This model, which is a slight extension of that presented in [7], is formulated by means of the following system of ordinary differential equations (ODEs):

$$\begin{aligned}
\frac{d\,x}{d\,t} &= (\lambda - k_1 - \beta - \mu)\,x + \lambda'\,(y_1 + y_2) - k_2 \frac{x(y_1 + y_2)}{x + y_1 + y_2} \\
\frac{d\,y_1}{d\,t} &= (-\mu - \beta')\,y_1 + k_2 \frac{x(y_1 + y_2)}{x + y_1 + y_2} \\
\frac{d\,y_2}{d\,t} &= k_1 x + (-\mu - \beta')\,y_2 \\
\frac{d\,z}{d\,t} &= \beta x + \beta'(y_1 + y_2) - \mu' z
\end{aligned} \tag{1}$$

where t is the time (years), x is the number of undetected HIV infected individuals, $(y_1 + y_2)$ is the known HIV infected population, and z is the number of AIDS patients, who are at the clinical, symptomatic stage. The known infected population is subdivided into y_1 and y_2 according to whether the individual infection was discovered by means of contact tracing (y_1) or by chance, a screening or any other way (y_2). This model contains several parameters, the most important ones being the rate of detection of HIV infection related to the contact tracing program (k_2), and the detection rate unrelated to the program (k_1). In this contribution, our main aim is the determination of the parameter values in the model (1), and the comparison of results with similar proposals [8].

In order to have a substantial series of test data, the model ODEs have been numerically solved, rather than using actual historical data. Previously, parameter values have been estimated from data, by means of statistical techniques. These values are shown in Table 1, together with either a confidence interval or the estimated standard deviation. For our purposes, considering the two distinct subpopulations y_1 and y_2 does not contribute towards parameter estimation, so we introduce the new state $y = y_1 + y_2$ and in the sequel we stick to the subse-

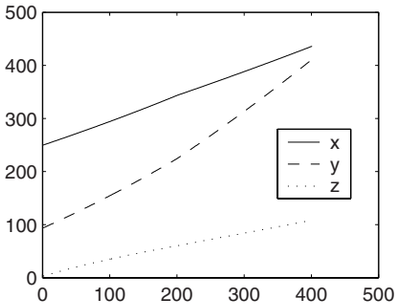


Fig. 1. System states generated for test data

quent model with only three equations. Starting from the initial values $x_0 = 250$, $y_0 = 94$, $z_0 = 3$, the model has been simulated from $t = 0$ to $t = 2$, obtaining a sequence of 200 values of the state vector. Then, some parameters have been modified as indicated in the last column of Table 1, and a new sequence of 200 states has been calculated, corresponding to the interval $t = [2, 4]$. This parameter change reflects an eventual improvement in the active search technique, as well as medical advances that could retard the development of symptoms and the mortality. The obtained data are represented in Fig. 1. Finally, gaussian noise with zero mean and variance 3, has been added to all obtained data, in order to simulate measurement errors when sampling a real situation.

Table 1. Actual parameters and their ranges

Parameter	Value, $t \in [0, 2]$	Interval	Std. Dev.	Value, $t \in [2, 4]$
λ	0.5594	[0.5353, 0.58494]		0.5628
μ	0.0053		0.00254	0.0053
μ'	0.76	[0.66, 0.85]		0.72
$b = \frac{1}{\beta}$	8.82		0.214	8.97
$b' = \frac{1}{\beta'}$	7.41		0.13	7.64
$r = \frac{\lambda'}{\lambda}$	0.0482		0.0295	0.0482
k_1	0.2161		0.0195	0.2328
k_2	0.2322		0.0096	0.2614

In order to simplify notation, it is convenient to cast the model (1) into the linearly parameterized form [3]:

$$\frac{d\,\boldsymbol{x}}{d\,t} = \boldsymbol{A}(\boldsymbol{x})\,\boldsymbol{\theta}\,; \quad \boldsymbol{A}(\boldsymbol{x}) = \begin{pmatrix} x & y & -\frac{xy}{x+y} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{xy}{x+y} & y & x & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & x & y & z \end{pmatrix}\,; \quad \boldsymbol{x} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{2}$$

where the vector $\boldsymbol{\theta}$ of formal parameters is defined in Table 2, which also shows the parameter values used for test data generation.

Table 2. Formal parameters

Parameter definition	Nominal value, $t \in [0, 2]$	Nominal value, $t \in [2, 4]$
$\theta_1 = \lambda - k_1 - \beta - \mu$	0.2246	0.2132
$\theta_2 = \lambda'$	0.0270	0.0271
$\theta_3 = k_2$	0.2322	0.2614
$\theta_4 = -\mu - \beta'$	-0.1403	-0.1362
$\theta_5 = k_1$	0.2161	0.2328
$\theta_6 = \beta$	0.1134	0.1115
$\theta_7 = \beta'$	0.1350	0.1309
$\theta_8 = -\mu'$	-0.7600	-0.7200

3 Parameter Estimation with Hopfield Neural Networks

Hopfield neural networks [9, 10] are recurrent systems that were shown to possess a Lyapunov function and, consequently, the stability of fixed points was proved. In the Abe formulation [11, 12], the evolution of a continuous Hopfield network is defined by the following system of ODEs:

$$\frac{du_i}{dt} = \sum_j w_{i\,j} s_j - I_i \; ; \qquad s_i(t) = \tanh\left(\frac{u_i(t)}{\beta}\right) \tag{3}$$

where s_i is the state of neuron i and β is a parameter to eventually control the slope of the hyperbolic tangent. Contrary to the original Hopfield formulation, Equation (3) has the advantage that the Lyapunov function has the simple form

$$V(\boldsymbol{s}) = -1/2 \sum_i \sum_j w_{i\,j}\, s_i\, s_j + \sum_i I_i\, s_i \tag{4}$$

where $w_{i\,j}$ and I_i are the network weights and biases, respectively. The procedure for the application of Hopfield networks to optimization [4] consists in the identification of the target function with the network Lyapunov function. For a particular target function, the weights and biases are then obtained, and the network can be implemented until it reaches a minimum of the target. Noticeably, the presence of the hyperbolic tangent forces the states to remain into the closed hypercube $\boldsymbol{s} \in [-1, 1]^n$. On one hand, this guarantees that states do not grow unboundedly. On the other hand, though, if the solution does not belong to this set, the network produces a wrong result. If the range of existence of the solution is a compact set, but it does not match the unit hypercube, it can be appropriately scaled or displaced, but the requirement persists that the compact interval that contains the solution must be known in advance.

The optimization methodology with Hopfield neural networks has been applied to parameter estimation [5] by searching a minimum of the prediction error. The network states $\boldsymbol{s}$ are a representation of the estimated parameters $\hat{\boldsymbol{\theta}}$ and, when the network reaches an equilibrium, the solution is contained in the state. For a linearly parameterized system given by Equation (2), the prediction error is $\boldsymbol{e} = \frac{d\,\boldsymbol{x}}{d\,t} - \boldsymbol{A}(\boldsymbol{x})\,\hat{\boldsymbol{\theta}}$, and the Lyapunov function is defined as $V = \frac{1}{2}\,\boldsymbol{e}^\top \boldsymbol{e}$. After

some algebra, and neglecting a constant term that does not change the position of the minima of V, the weight matrix and the bias vector are obtained: $\boldsymbol{W} = -\boldsymbol{A}^T \boldsymbol{A}$, $\boldsymbol{I} = -\boldsymbol{A}^T \frac{d\boldsymbol{x}}{dt}$. Since the matrix $\boldsymbol{A}$ depends on the system state $\boldsymbol{x}$, the weights vary with time and, strictly speaking, the stability of the network is not guaranteed. However, optimization is still achieved as long as the system evolution is not too fast. In order to apply this technique to the HIV model described in Sect. 2, some implementation details must be solved. First of all, as mentioned in the above paragraph, network states can not escape the closed unit hypercube. Since probable parameter values belong to this set (Table 2) this requirement is fulfilled in this case. Otherwise, a displacement via the definition of *nominal values* would be necessary [5], with no loss of generality. A more severe issue became apparent during preliminary simulations: the continuous model (2) is not a valid approximation when only a set of discrete data observations is available. Instead, a discrete model must be explicitly formulated:

$$\boldsymbol{x}_{n+1} = \boldsymbol{x_n} + \Delta t \, \boldsymbol{A}(\boldsymbol{x_n}) \, \boldsymbol{\theta} \tag{5}$$

which is simply the solution of the model ODE by the Euler rule. Since we have a data sequence of 400 values for the time interval $t \in [0, 4]$, the step size must be fixed as $\Delta t = 0.01$. Thus, the prediction error at time $n+1$ is $\boldsymbol{e}_{n+1} = \boldsymbol{x}_{n+1} - \boldsymbol{x_n} - \Delta t \, \boldsymbol{A}(\boldsymbol{x_n}) \, \hat{\boldsymbol{\theta}}$, so that identifying the Lyapunov function $V = \frac{1}{2} \boldsymbol{e}^\top \boldsymbol{e}$ with the standard form (4) yields the weights and biases:

$$\boldsymbol{W} = -\Delta t^2 \, \boldsymbol{A}^T \boldsymbol{A} \qquad \boldsymbol{I} = -\Delta t \, \boldsymbol{A}^T \left(\boldsymbol{x}_{n+1} - \boldsymbol{x_n} \right) \tag{6}$$

where, as in the continuous case, a constant has been neglected.

4 Simulation Results and Discussion

In this section, the proposed method is applied to the HIV modelling problem. Initially, the data used in the simulated experiment is the sequence obtained before the noise was added. All states are assigned initial values with a 30 % relative error over the correct values (see Table 2). The network equations (3) are numerically solved, with the same choice of the step size as in data generation: $\Delta t = 0.01$. The slope of the hyperbolic tangent is identically set for all neurons at $\beta = 1$. The prediction error $\boldsymbol{e}$ is shown in Fig. 2.a). It is apparent that, when the network reaches a stationary state, approximately at $t = 0.5$, the error approaches zero. Also, after the sudden parameter change at $t = 2$, the network quickly settles again. The squared error, averaged along the complete simulation, is $\sum_i \boldsymbol{e}_i^\top \boldsymbol{e}_i = 0.7869$. We conclude that the network presents fast convergence and quickly adapts itself when sudden parameter changes occur, so that it satisfactorily models the system, with regard to output prediction.

Concerning parameter estimation, as an example, in Fig. 2.b) the estimation of the fourth component of the parameter vector $\boldsymbol{\theta}$ is shown, together with its real value, evidencing a large relative estimation error. In the following, we try to explain this apparent contradiction -accurate prediction with poor estimates-

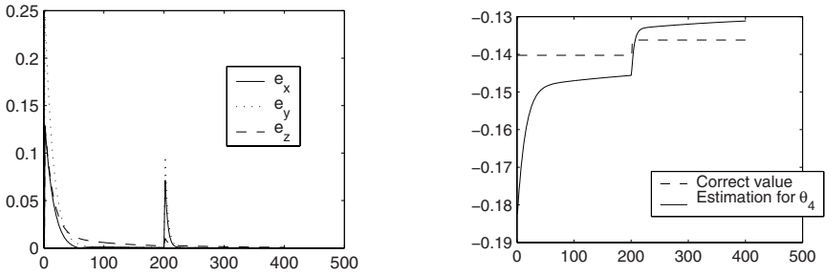


Fig. 2. a) Absolute state prediction error. b) Estimation of θ_4 and real value

as well as offer ideas for its solution. Firstly, it is well known in the control engineering literature [3, 1] that if the system dynamics is not complex enough, many sets of incorrect parameter values can lead to a correct prediction. This is a common drawback of all estimation methods, since they are driven by the prediction error. It is obvious from Fig. 1 that the system is not particularly complex. In physical systems that have inputs, this problem can be alleviated by providing *persistent excitation*: feeding the system with a rich input signal causes the output to become so complex that the only way to cancel the prediction error is to produce an accurate parameter estimation. Since the HIV-AIDS model we are dealing with has no inputs, this idea offers no solution but it sheds some light on the problem: there are "too many" parameter values that produce correct prediction partly because there are "too many" parameters to estimate. Some of the parameters -e.g. the death rates- can be accurately estimated from global statistics, while the detection rates k_1 and k_2 are, not only more difficult to obtain, but also specially important because they provide a measure of the efficiency of the contact tracing program. We are currently adapting our network to estimate a smaller number of parameters and we expect better results. Another aspect of the implementation is the different ranges of the columns of the matrix $\boldsymbol{A}$. For instance, the first column of $\boldsymbol{A}$ is two orders of magnitude larger than its last column, so the convergence of θ_8 is much slower than that of θ_1. The choice of diverse values of β for each neuron could contribute, but this adjustment is rather problematic because instabilities may appear if β is too small. Finally, simulations were repeated using the noisy data patterns and the results were even worse. We conjecture that the relation between every two consecutive states -that appears in the expression of the weights- is completely destroyed by the high frequency noise. In order to deal with this extreme sensitivity to noise, some smoothing preprocess should be applied to data.

5 Conclusions and Future Directions

In this contribution, the optimization methodology of Hopfield and Tank is adapted to deal with parameter estimation, in the context of dynamical system

identification. The resulting network has a dynamic nature, since its weights are time-varying. The proposed technique has been applied to adjusting the parameters of a model of the Cuban HIV-AIDS epidemics. Simulation results show that the network achieves an accurate output prediction, which could be used for infection forecasting. The fast convergence means that a reduced number of data patterns are needed before the prediction error approaches zero. Moreover, after a sudden parameter change the network stabilized itself rather quickly. Although exhaustive tests of computational cost were not performed, no special attention was paid to implementation enhancement -in fact, an interpreted language was used- and response times were reasonable. The main weaknesses of the proposed method were a poor estimation of numeric values of parameters, as well as excessive sensitivity to noise. While the latter problem can be alleviated by smoothing the data, the former was probably due to the excessive number of estimated parameters, when compared to the number of measured states. As a summary, we can affirm that the neural estimator offers promising results in on-line identification of systems with varying parameters and a large number of observable states for which noiseless measures are available at a high frequency, such as robotic systems. Further research is currently being developed in order to improve the estimation results in "hard" systems, such as the HIV-AIDS model.

References

1. Ljung, L..: System Identification. Theory for the User. Prentice Hall (1999)
2. Unbehauen, H.: Some new trends in identification and modeling of nonlinear dynamical systems. Applied Mathematics and Computation **78** (1996) 279–297
3. Slotine, J.J., Li, W.: Applied Nonlinear Control. Prentice Hall (1991)
4. Tank, D., Hopfield, J.: 'Neural' computation of decisions in optimization problems. Biological Cybernetics **52** (1985) 141–152
5. Atencia, M.A., Joya, G.: Gray box identification with Hopfield neural networks. Revista Investigacion Operacional (accepted for publication) (2003)
6. Bailey, N.T.: The mathematical theory of infectious diseases and its applications. Ch. Griffin and Company LTD (1975)
7. de Arazoza, H., Lounes, R.: A non-linear model for a sexually transmitted disease with contact tracing. Mathematical Medicine and Biology: A Journal of the IMA **19** (2002) 221–234
8. Pedroso-Rodríguez, L.M.: A genetic algorithm based solution for the parameter estimation in models defined by means of ordinary differential equations, Diploma Thesis (original version in Spanish) (2002)
9. Hopfield, J.: Neural networks and physical systems with emergent collective computational abilities. Proc. Natl. Acad. Sci. USA **79** (1982) 2554–2558
10. Hopfield, J.: Neurons with graded response have collective computational properties like those of two-state neurons. Proc. Natl. Acad. Sci. USA **81** (1984) 3088–3092
11. Abe, S.: Theories on the Hopfield neural networks. In: Proc. IEE International Joint Conference on Neural Networks. Volume I. (1989) 557–564
12. Joya, G., Atencia, M.A., Sandoval, F.: Hopfield neural networks for optimization: Study of the different dynamics. Neurocomputing **43** (2002) 219–237

Comparison of Neural Models, Off-line and On-line Learning Algorithms for a Benchmark Problem

António E. B. Ruano

CSI, FCT, University of Algarve, 8000 Faro, Portugal

aruano@ualg.pt

Abstract. This paper compares the application of different neural models – multilayer perceptrons, radial basis functions and B-splines - for a benchmark problem, and illustrates the applicability of a common learning algorithm for all models considered. The learning algorithm is employed both for off-line training and for on-line model adaptation. In the latter case, a sliding window of past learning data is employed.

1 Introduction

Neural networks and neuro-fuzzy systems are now established models for systems identification. This paper compares the performance of different models and training algorithms for a specific problem. The problem, the evolution of HIV and AIDS in Cuba, is described in Section 2. Section 3 introduces the models employed, and describes the performance of these models, subject to off-line training. The adaptation problem is reported in Section 4. Final conclusions are drawn in Section 5.

2 The problem

The data used in this work tries to model a real problem: the evolution of the HIV and AIDS epidemics in Cuba. In a certain moment, the health service in this country employed an active search of the persons possibly infected with HIV, as a result of interviews conducted with people where AIDS has been identified. This policy produced its benefits, and the rate of increase of people with AIDS has been decreasing since then. The data that is used here is not the original historical data, but it was generated from model (1), with parameters tuned within realistic ranges.

$$\begin{aligned}
\frac{dX}{dt} &= (\lambda - k_1 - \beta - \mu)X + \lambda'(Y_1 + Y_2) - k_2 \frac{X(Y_1 + Y_2)}{X + Y_1 + Y_2} \\
\frac{dY_1}{dt} &= (-\mu - \beta')Y_1 + k_2 \frac{X(Y_1 + Y_2)}{X + Y_1 + Y_2} \\
\frac{dY_2}{dt} &= (-\mu - \beta')Y_2 + k_1 X \\
\frac{dZ}{dt} &= \beta X + \beta'(Y_1 + Y_2) - \mu' Z
\end{aligned} \tag{1}$$

In (1), Z denotes the numbers of persons with AIDS, X is the number of persons with HIV that have not been detected, $Y=Y_1+Y_2$ represents the number of persons with HIV that have been identified, with Y_1 representing the ones detected as a result of Cuba's

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 457-464, 2003.

active search policy, and Y_2 the ones identified by pure chance. To generate the data, before active search was in place, the following parameters were used:

$$\lambda = 0.5594, \mu = 0.0053, \mu' = 0.76, \beta = 1/b\,(b = 8.82), \beta' = 1/b'\,(b' = 7.41)$$
$$\lambda' = r/\lambda\,(r = 0.0482), k_1 = 0.2161, k_2 = 0.2322 \quad (2)$$

with initial values $X(0) = 250, Y_1(0) = 26, Y_2(0) = 68, Z(0) = 3$.

At time instant 201, the parameters changed to the ones indicated below:

$$\lambda = 0.5628, \mu = 0.0053, \mu' = 0.72, \beta = 1/b\,(b = 8.97), \beta' = 1/b'\,(b' = 7.64)$$
$$\lambda' = r/\lambda\,(r = 0.0482), k_1 = 0.2328, k_2 = 0.2614 \quad (3)$$

In total, 400 data points have been generated using this model. This data has been subsequently contaminated with Gaussian noise (0, 3). The first 150 data points are employed in this work for training and the next 50 for testing. The last 200 points (where the dynamics have changed) will be used for adaptation purposes.

Obviously, Z is admitted to be unknown. Three different models will be derived:

$$N[k] = f_i(Y_1[k-1], Y_2[k-1], Z[k-1]), \quad N[k] = Y_1[k], Y_2[k], Z[k] \quad (4)$$

The model inputs have been selected as a result of the *a priori* knowledge of the function (1) generating the data. Although filtering the data is an action that could (should) be performed, it was not done here, in order to assess the potentials of neural networks to determine the actual function behind the data. It should be noted that this is not a typical modelling problem, as the range of the test and adaptation data lie outside the range of the training data (see figure 1). It is therefore an extrapolation problem, with the dynamics of the data changing in the adaptation set.

3 Off-line training

In this section, results obtained with models trained off-line are compared. The first 149 samples are used for training and the next 50 samples for testing. The last 200 samples will be used for adaptation. For all models, their performance is evaluated using two criteria: the **M**ean-**S**quare **E**rror (MSE=$\frac{1}{m}\sum_{i=1}^{m} e^2[i]$) and the **M**aximum **A**bsolute **E**rror (MAE=$\max_{i\in[1\ldots m]} |e[i]|$), over the training, test, and adaptation sets.

3.1 Linear Models

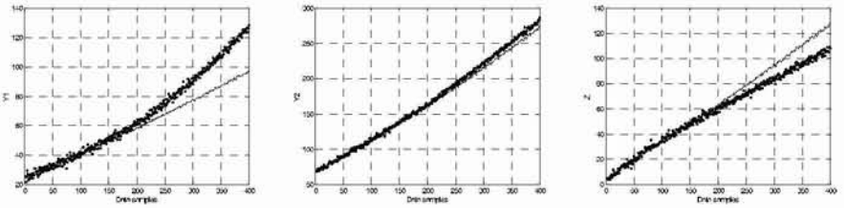
The first models employed are linear models:

$$N[k] = [1 \quad Y_1[k-1] \quad Y_2[k-1] \quad Z[k-1]]\,\mathbf{w},\ N[k] = Y_1[k], Y_2[k], Z[k] \quad (5)$$

The complexity of the linear models is 4. As it can be seen in figure 1, the dynamics change from sample 200 onwards. As the parameters of the linear models are estimated with data with the previous dynamics, the estimation is not reliable for the latter period. The results obtained with this model type are summarized in Table 1.

3.2 Multilayer Perceptrons

Multilayer perceptrons (MLPs) with sigmoid functions in the hidden layers and a linear activation function in the output layer are employed here. In terms of mathematical model, they can be described as:



a) Y_1 b) Y_2 c) Z

Fig. 1. Results obtained with linear models. Solid line: Predicted output; dotted line: Real data.

Table 1. Results obtained with linear models

	Y_1	Y_2	Z
MSE (training)	3.15	6.26	3.42
MAE (training)	7.03	8.41	6.66
MSE (test)	6.06	7.48	9.45
MAE (test)	5.01	5.78	7.60
MSE (ada)	279.05	52.02	150.16
MAE (ada)	31.05	15.55	22.69

$$y = \sum_{i=0}^{M} w_i g_i(\mathbf{x}, \mathbf{v}_i) \tag{6}$$

In (6), $\mathbf{x} = \left[Y_1[k-1] \quad Y_2[k-1] \quad Z[k-1] \right]$, $g_0(.,.) = 1$, and $\mathbf{v}$ denotes the parameters which appear nonlinearly in the approximator output. The results shown in the tables below are obtained with MLPs belonging to 2 different topologies: the first three columns to a topology [4 1], and the last three columns to [2 2 1]. In terms of complexity, the former has 16 nonlinear parameters and 5 linear parameters (21 in total) and the latter has 14 nonlinear parameters and 3 linear parameters (17 in total). In what concerns the training procedure, the initial values of the weights associated with the neurons in the hidden layer(s) were initialised with a heuristic ensuring that all neurons were in an active region, and that the condition of the initial Jacobean matrix was not exacerbated. The training algorithm employed was the Levenberg-Marquardt algorithm, minimizing a training criterion that exploits the separability of the linear-non-linear parameters (for more details please see [1]). Training was terminated, whether when a minimum was found, according to the termination criteria (7) (with $\tau_f = 10^{-3}$), or when a minimum in the MSE of the test data was identified.

$$\begin{aligned} &\Omega(k-1) - \Omega(k) < \theta(k) \\ &\left\| \boldsymbol{\lambda}(k-1) - \boldsymbol{\lambda}(k) \right\| < \sqrt{\tau_f} \left(1 + \left\| \boldsymbol{\lambda}(k) \right\| \right) \\ &\left\| \mathbf{g}(k) \right\| \leq \sqrt[3]{\tau_f} \left(1 + \Omega(k) \right) \end{aligned} \tag{7}$$

In (7), Ω denotes the training criterion (the sum of the squares of the errors), $\boldsymbol{\lambda}$ the network weight vector associated with the nonlinear parameters, $\mathbf{g}$ the gradient and k

the current iteration, $\theta(k) = \tau_f \left(1+\Omega(k)\right)$, $\| \ \|$ is the 2-norm and τ_f is a measure of the desired number of correct figures in the training criterion.

Table 2. Results obtained with MLPs - Y_1

	1	2	3	1	2	3
MSE (training)	2.55	2.59	2.67	2.92	2.70	2.83
Max Error (training)	6.11	6.39	6.22	6.39	6.55	6.11
MSE (test)	12.88	13.01	7.41	3.94	7.66	3.83
Max Error (test)	8.05	7.86	6.52	4.25	5.94	4.24
MSE (val)	2,092	1,691	1,426	1,293	1,433	1,374
Max Error (val)	31.04	73.10	71.38	69.53	68.82	67.96

Table 3. Results obtained with MLPs – Y_2

	1	2	3	1	2	3
MSE (training)	5.49	5.32	5.75	5.86	5.82	5.92
Max Error (training)	7.42	7.36	7.47	7.63	7.62	7.64
MSE (test)	48.81	38.92	14.88	4.38	6.75	10.42
Max Error (test)	15.55	12.69	9.09	4.51	6.49	6.68
MSE (val)	6,256	5,485	4,037	2,876	4,045	1,785
Max Error (val)	135.14	129.28	119.41	104.59	123.97	84.36

Table 4. Results obtained with MLPs – Z

	1	2	3	1	2	3
MSE (training)	3.02	2.92	2.97	2.90	2.80	3.01
Max Error (training)	6.42	5.85	5.57	6.78	6.84	6.34
MSE (test)	16.24	9.19	5.03	13.59	8.24	9.90
Max Error (test)	8.16	6.21	5.36	8.30	6.41	6.32
MSE (val)	1,026	891	900	910	670	731
Max Error (val)	54.22	51,81	52.04	51.63	46.61	48.07

3.3 Radial Basis Functions

Gaussian activation functions are used in the hidden layers, while a linear activation function is employed in the output neuron. Mathematically, these models can be described as:

$$y_n = \sum_{i=1}^{M} w_i \Phi_i(\mathbf{x}, \mathbf{x}_i), \quad \Phi_i(\mathbf{x}, \mathbf{x}_i) = e^{-\frac{1}{2\sigma_i^2}\|\mathbf{x}-\mathbf{x}_i\|} \tag{8}$$

The weighting functions Φ_i possess local behaviour, around the operating point $\boldsymbol{x}_i$. This means that the value of the weighting functions decreases with an increasing distance from the input vector $\boldsymbol{x}$ to the operating point $\boldsymbol{x}_i$. Please notice that here no bias is employed in the output neuron. The initial values of the Gaussian activation functions were determined employing an adaptive version of the *k-means clustering* algorithm [2]. As in the case of MLPs, training was performed using the Levenberg-Marquardt algorithm, minimizing a training criterion that exploits the separability of the linear-non-linear parameters [3]. The same termination criteria employed in the

case of MLPs was used. The results shown in the table below are obtained with RBFs with 4 hidden neurons. In terms of complexity, this model has 16 nonlinear parameters and 4 linear parameters (20 in total).

Table 5. Results obtained with RBFs

	Y_1	Y_2	Z
MSE (training)	2.81	6.17	3.14
Max Error (training)	6.70	7.94	6.54
MSE (test)	7.68	4.54	9.4
Max Error (test)	5.06	4.76	7.29
MSE (val)	210.57	47.74	2,667
Max Error (val)	43.39	20.21	95.4

3.4 B-Spines

B-splines networks are a kind of Associative Memory Networks which differ from other neural networks because they store the information locally ("learning" about one part of the input space minimally affects the rest of it). They can be seen as neural networks, or fuzzy models. If certain assumptions are met, B-spline networks are strictly equivalent to Mamdani fuzzy models [4, 5].

B-spline networks are local models, which decompose the operational range into different operating regimes, a model being developed for each region. The overall model output is given by the combination of all locally active submodels:

$$y_n = \sum_{i=1}^{M} g_i(\mathbf{x}, w_i)\Phi_i(\mathbf{x}, \mathbf{v}_i) \tag{9}$$

Each sub-model g_i is valid in a region specified by its corresponding weighting function Φ_i. The weighting functions describe a partition of the entire operational range into subregions, and they determine the kind of transiction between neighboring submodels.

The design of a B-spline network is a complex process, and usually constructive algorithms are used. In this paper, we compare the performance of a heuristic approach, the ASMOD algorithm [6], which minimizes the **B**ayesian **I**nformation **C**riterion (**BIC**):

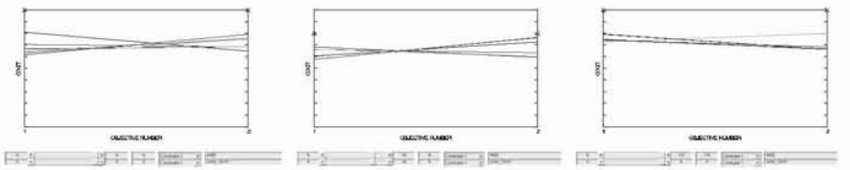
$$BIC = m\ln(J) + p\ln(m) \tag{10}$$

with a **M**ulti-**O**bjective **G**enetic **P**rogramming approach (MOGP) [7]. In (10), m is the number of training patterns, J is the MSE of the model and p is its complexity. The ASMOD algorithm is an iterative technique, where in each iteration a set of candidates is generated. The ASMOD algorithm terminates either when the best model within the set of generated candidates has a greater BIC than current best one, or when the MSE of the test set of the current best model is lower than the one obtained with the best model in the current iteration.

Regarding the MOGP approach, one objective, the MSE for the training set, and one restriction, the MSE for the test set, were considered. The goals for the two criteria were chosen according to the performance obtained for the models previously considered. The size of the population employed was 20, and the algorithm run for 20

generations. The method used for the creation of the initial population was *ramped half-and-half*, *stochastic universal sampling* was employed, an *exponential rank-based fitness assignment* with a value of *selective pressure* of 5 was used, and the *mutation* and *crossover probabilities* employed had the values 0.7 and 0.6, respectively [8].

The graphs in figure 2 show the non-dominated solutions found for each model. The cross in each axis indicates the goal assigned for each criterion. As it can be noticed, there are several solutions that have been found by the algorithm, and it is up to the designer to select one.



a) Y_1 b) Y_2 c) Z

Fig. 2. Non-dominated solutions obtained using the MOGP algorithm

The results obtained with B-spline networks are summarized in table 6. The first 3 columns show the results obtained with the ASMOD algorithm, and the latter three the results achieved with one selected solution of the MOGP technique.

Table 6. Results obtained with B-splines: ASMOD-first columns; MOPG – last columns

	Y_1	Y_2	Z	Y_1	Y_2	Z
MSE(train)	2.98	6.06	3.06	2.45	5.76	2.55
MAE(train)	6.65	8.17	6.93	5.50	8.61	5.26
MSE (test)	17.78	5.18	3.63	3.16	7.62	4.78
MAE (test)	11.58	6.39	4.41	3.97	6.21	5.47
MSE (ada)	22,046	9,964	949	154	288	4.17
MAE (ada)	311	222.8	69.34	25.38	40.44	5.13
Complexity	(1,9)	(1,8)	(1,9)	(24,30)	(15,21)	(16,20)
	(10,18)	(11,18)	(11,19)	(33,39)	(26,32)	(24,28)

In this table, the pair (x,y) in the last row denotes the number of nonlinear parameters (interior knots) – x, and the number of linear parameters – y – of the model. This type of model has an additional complication. As the basis functions have compact support, the network is only valid within a specified range of its inputs. As in this problem the samples in the adaptation set are not within the range of the training and test sets, in order to assess the performance of the network in the adaptation set, the range of usability of the model must be extended. Here it will be assumed that nothing is known about the adaptation set, apart from its range. In this context, a sensible strategy is to extend the range of the polynomial active for the last internal interval so that this additional range is accounted for. This can be done in several ways. It is possible only to enlarge the last internal interval in order to accommodate the whole operational range, or to introduce additional interior knots within the range of the adaptation data. The solution employed here is the last one. A number of interior

knots is introduced within the adaptation range, so that the interval length, there within, is the same as the length of the last interval, within the original range. The additional weights must be determined in such a way that the polynomials obtained for the additional intervals are the same as in the last original internal interval.
The second pair in the last row of table 6 illustrates the complexity of the model, extended for the additional range, in the way described above.

3.5 Model Comparison

With the results expressed in tables 1 to 6, the following conclusions can be taken:

a) Linear model and neural models:
   - ✓ All neural models obtain better results in the training set;
   - ✓ There is no clear evidence than linear models are best or worse than nonlinear models, in the test set;
   - ✓ MLPs are worse than the linear models in the adaptation set; RBFs and B-splines are in two cases better than linear models;
   - ✓ Linear models have a substantially smaller complexity than nonlinear models;

b) Neural models:
   - ✓ In what concerns MLPs, the two-hidden layer topology achieves, in overall, better results than one-hidden layer networks (despite the smaller complexity of the former);
   - ✓ The performance of RBFs is almost equivalent to the 2-hidden layer MLPs, in the training and test sets; the performance of MLPs in the adaptation set is worse than RBFs; the complexity of the two models is equivalent;
   - ✓ The performance of B-splines, trained with the MOGP approach is significantly better than the ones trained with ASMOD, for all cases;
   - ✓ On overall, the performance of the B-splines is better than RBFs, and best overall, at the expense of being the most complex model.

4 Adaptation

The models obtained in the previous section were adapted on-line, employing the last 200 samples of data. An on-line version of the LM algorithm, employed for off-line purposes, is used here. The algorithm is employed at every sample instant, for a sliding window (SW) of past data. The initial values of the nonlinear parameters are the ones obtained as the result of the previous execution of the algorithm. The initial regularization parameter is the final value of this parameter in the previous execution. Each execution terminates either when either the termination criteria (7) are met, or a predefined number of iterations is achieved. For MLPs and RBFs a 50 samples SW is used, while for B-splines a 80 samples SW is employed.
The performance of the models is evaluated using the same criteria as in the last section, but applied to different data. At each sample instant, the predicted output ($\hat{y}[k]$) is computed, and afterwards the model is updated, in the manner described above. The model output is computed again (the output *a posterior*- $\overline{y}[k]$), and all actions are repeated in the next sample time. In the two first rows, the MSE and the MAE of the predicted output are shown. The next two rows compute the same criteria, but this time for the *a posterior* output. The last two rows evaluate how the

model is distorted, as a result of on-line adaptation. After the on-line algorithm is applied to the whole adaptation set, the resulting models are employed to the training and test sets. If the model did not forget what it was (off-line) trained for, as a result of on-line adaptation, then the last two rows should present value similar to the ones obtained in the first 4 rows of tables 2-6.

Table 7. Results of on-line adaptation for all neural models

	Y_1	Y_2	Z	Y_1	Y_2	Z	Y_1	Y_2	Z
MSE($\hat{y}[k]$)	4.86	7.14	3.30	5.58	12	4.55	6.93	9.29	6.39
MAE($\hat{y}[k]$)	8.87	8.09	4.66	6.27	22.83	7.73	12.75	20.33	14.6
MSE($\overline{y}[k]$)	4.81	4.10	2.49	4.35	5.32	3.36	2.18	3.36	2.14
MAE($\overline{y}[k]$)	5.06	5.77	4.27	5.74	7.2	4.57	4.19	5.18	5.29
MSE (tra+t)	2,090	11,366	3,541	$35*10^6$	$5.1*10^5$	4,576	751.8	1,408	133
MAE(tra+t)	73.73	443.33	341.35	12,285	1,050	79,63	15.9	184.8	70.98

It can be seen that the performance of MLPs (first 3 columns) and RBFs (next 3) is similar. B-splines (last 3) obtain worse results in the predicted output, but better in the *a posterior* output. As expected, B-splines achieve less distortion.

5 Conclusions

This is not the best of the problems for neural models. They should be used for interpolation (approximation) and not for extrapolation purposes. Additionally, as the data employed is nearly linear, there is not much to gain in the use of nonlinear models, compared with simpler, linear models.

On the overall, B-splines perform slightly better than MLPs and RBFs, for the performance criteria employed. The learning algorithms, both off-line and on-line, obtain good models. MOGP achieves better B-spline models than the ASMOD algorithm.

References

1. Ruano, A.E., Ferreira, P.M., Cabrita, C., Matos, S.: Training Neural Networks and Neuro-Fuzzy Systems: a Unified View, 15th IFAC World Congress, Barcelona, Spain, (2002)
2. Chinrungrueng, C., Séquin, C.H.: Optimal adaptive k-means algorithm with dynamic adjustment of learning rate, IEEE Transactions on Neural Networks, Vol. 6, N° 1 (1995) 157-169
3. Ferreira, P.M., Ruano, A.E.: Neural Network Models in Greenhouse Environmental Control, NeuroComputing, Vol. 43, N° 1 (2002) 51-75
4. M. Brown, Harris C.: Neurofuzzy adaptive modelling and control, Prentice Hall, London (1994)
5. Ruano, A.E., Cabrita, C., Oliveira, J.V., Kóczy, L.T.: Supervised Training Algorithms for B-Spline Neural Networks and Neuro-Fuzzy Systems, International Journal of Systems Science, Vol. 33, N° 8 (2002) 689-711
6. Weyer, E., Kavli, T.: The ASMOD algorithm: some new theoretical and experimental results, Technical Report (1995)
7. C. Cabrita, Ruano A. E., Fonseca, C.M.: Single and multi-objective genetic programming design for B-spline neural networks and neuro-fuzzy systems, Proc. IFAC Workshop on Advanced Fuzzy/Neural Control 2001 (AFNC'01), Valencia, Spain (2001) 93-98
8. J.R. Koza: Genetic Programming: On the programming of computers by means of natural selection, MIT Press (1992)

Using Neural Networks in a Parallel Adaptative Algorithm for the System Identification Optimization

Juan A. Gómez Pulido, Juan M. Sánchez Pérez, Miguel A. Vega Rodríguez

Departamento de Informática. Escuela Politécnica, Campus Universitario s/n,
10071 Cáceres, Spain
{Jangomez, Sanperez, Mavega }@Unex.es
http://arco.unex.es

Abstract. In this work[1] we present the use of neural networks to implement processing units of a parallel adaptative algorithm for high precision system identification. The proposed algorithm uses recursive least squares processing and ARMAX modeling. After explaining the algorithm and the tunning of its parameters, we show the system identification for four benchmarks with different implementations of this algorithm, demonstrating how neural networks improve the result precision.

1 Introduction

1.1. System Identification

System Identification (SI) [1] tries to find a parametric model of dynamical systems from its input and output (I/O) measured values. In this work we consider "single input single output" (SISO) sampled systems with period T and parametric polinomial ARMAX modelation [2]:

$$A(q)\, y(k) = B(q)\, u(k\text{-}nk) \qquad (1)$$

being

- $A(q) = 1 + a_1 q^{-1} + ... + a_{na} q^{-na}$, the a_i represent the output parameters (size na)
- $B(q) = b_1 + b_2 q^{-1} + ... + b_{nb} q^{-nb+1}$, the b_i represent the input parameters (size nb)
- nk, output-input delay; q, delay unit (q^{-p} delays x(k) to x(k-p)); na ≥ nb

SI tries to determine the ARMAX model parameters (a_i, b_j) from measured I/O. Then, it is possible to compute the estimated output $y_e(k)$ and compare it with the real output y(k), computing the generated error (Fig.1):

$$y_e(k) = [-a_1 \cdot y(k\text{-}1) - ... - a_{na} \cdot y(k\text{-}na)] + [b_1 \cdot u(k\text{-}nk) + ... + b_{nb} \cdot u(k\text{-}nk\text{-}nb+1)] \qquad (2)$$

---

[1] This work has been developed thanks to TRACER project, TIC2002-04498-C05-01

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 465-472, 2003.

where

- $y_e(k)$ is the estimated output by the model
- y(k), u(k) are real output and input at present time
- y(k-1),..., u(k-1),... are real outputs and inputs at previous time

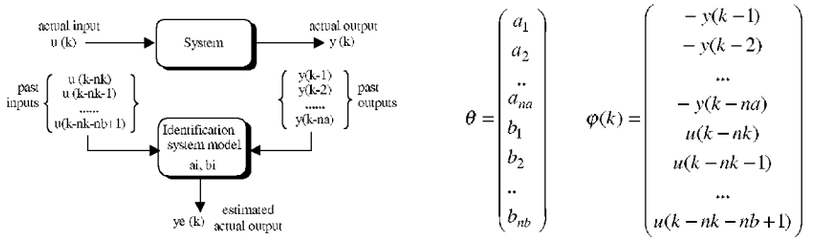


Fig. 1. System identification algorithm based on parametric estimation: inputs and outputs are measured and computed to generate the estimated output from an ARMAX model. The estimated output, is $y_e(k) = \varphi^T(k) \cdot \theta$, where θ is the parameter vector and $\varphi(k)$ is the data vector.

1.2. Identification modes

The recursive estimation updates θ in each time k, modelling so the system. To more sampled data processed, better precision for the model because it has more information about the system behaviour history. We consider SI performed by the well known Recursive Least Squares (RLS) with forgetting factor (λ) algorithm [3]. From the initial conditions, we start building $\varphi(k)$, then:

Initial conditions: k = p, $\theta(p)=0$, $P(p)=1000 \cdot I$, where I is the identity matrix and p the initial time (3)

$$y_e(k)=\varphi^T(k)\cdot\theta(k-1) \rightarrow \text{err}(k)=(y(k)-y_e(k)) \rightarrow K=\frac{P(k-1)\cdot\varphi(k)}{\lambda+\varphi^T(k)\cdot P(k-1)\cdot\varphi(k)}$$

$$P(k)=\frac{P(k-1)-K\cdot\varphi^T(k)\cdot P(k-1)}{\lambda} \rightarrow \theta(k)=\theta(k-1)+K\cdot\text{err}(k)$$

This algorithm is specified with λ (forgetting factor), initial values and the observed I/O {u(k), y(k)}. There is not any fixed value for λ, even it is used a value between 0.97 and 0.995 [4]. The cost function **F** is defined as the value to minimize:

$$F(\lambda)=\sum_{k=k_0}^{k=k_0+SN-1}\left|y_e(k)-y(k)\right| \qquad \text{(SN is the sample number)} \tag{4}$$

The recursive identification is very useful for predicting the system behaviour when there is a high degree of complexity and variability in the response. So, it is necessary

to elaborate, using SI, a mathematical model ($\theta(k)$) for covering the system behaviour under any type of input. As identification advances in the time, the predictions improve using more precise models. For example, we can compute in sample time the system model and then, with this model to simulate the system future behaviour, forwarding real situations (Fig. 2).

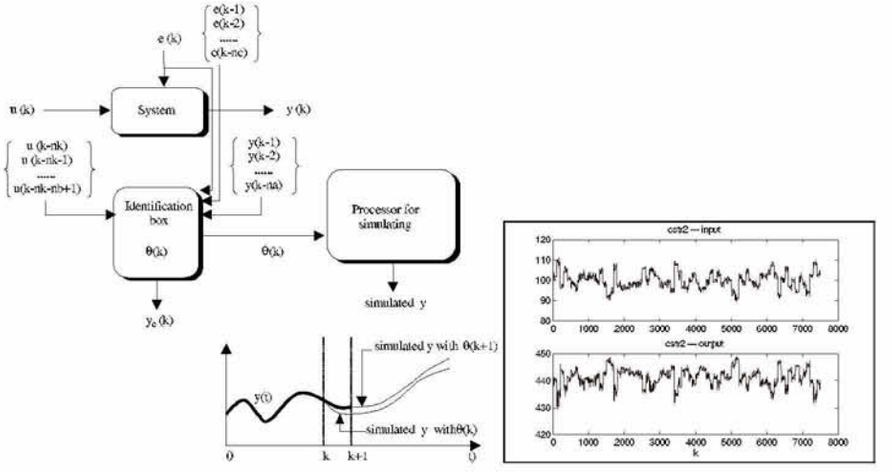


Fig. 2. Recursive SI allows us to predict and simulate future system behaviour. On the right side, a capture of a real SISO system benchmark used.

2. A parallel adaptative algorithm with neural networks to improve SI precision

We are interested in the system behaviour prediction in running time, that is, while the system is working and its I/O are being observed. At the same time, our first effort is to obtain high model precision (minimal F). SI precision is due to several causes, mainly to the forgetting factor λ. There is not any determined value for λ, but it is usually used within 0.995 and 0.98 [2][4]. Frequently this value is critical for model precision. Other sources also can have certain degree of influence (dimensions, initial values, the system...). Also, it may appear the precision problem when the system response changes quickly. Then the sample frequency must be high for avoiding the key data lost in the system behaviour description, and this implies a computational cost. We find the trade-off between a high sample frequency and a high precision in the algorithm computation. We enhance the required precision by λ using a parallel adaptative algorithm with neural networks, and we are actually reducing its computational cost designing especialized processors with reconfigurable hardware [10].

Fig. 3 shows the estimated output for 2 λ values for the *ball* benchmark [5] and the ARMAX model with na=3, nb=2, nk=1 running RLS. The estimated output is quite different according to the λ value. Making computations for 155 values of λ in the range 0.6 – 1.2 we find as optimum value λ = 0.98961 (F = 1.40762). This set of experiments is not possible to perform in real time systems. Moreover, 0.98961 is an optimum λ only for *ball* with those dimensions, because each system has its optimum values, so an unique λ for all SI may drive to a precision lost in many cases. By that, it is interesting to research about algorithms for finding the best λ values for each system. In relation with the model dimensions, we can say [6] that for a dimension of reasonable computational cost, λ is the most important optimization parameter.

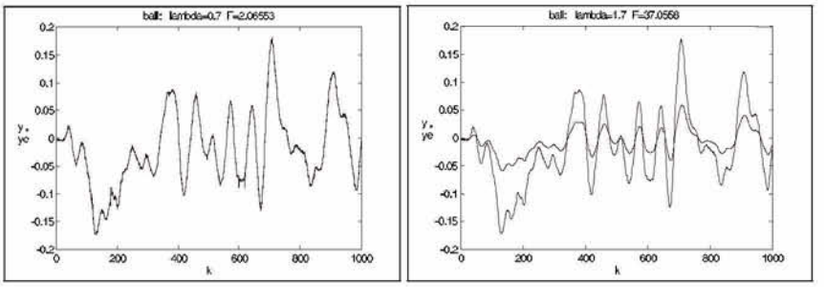


Fig. 3. System Identification RLS of benchmark *ball* with different values for λ parameter.

For finding the optimum λ, we propose PARLS (*Parallel Adaptative Recursive Least Squares*) algorithm. Starting of an initial λ (λc) and an initial R (the interval of generation where λc is in the middle), a set of λ is generated covering uniformly all the R interval. The λ values generated are equal to the number of parallel processing units. Each phase of PARLS loop corresponds to a given number of sample times (PHS) running SI with the considered λ. We use the following nomenclature:

R	generation interval
λc	λ central in R
PHS	phase samples

PUN	number of parallel processing units
TSN	total number of samples
RED	reduction factor of R

Table 1. PARLS nomenclature.

In each phase, R is reduced by the RED factor, in such a way that the generated set of λ will be more and more near of the previous optimum found. In each processing unit, during each phase, the cost function F is computed (F is defined as the accumulated error of the samples that constitutes each phase). From eq. (4), we have:

$$F(\lambda_{PUx}) = \sum_{k=k_0}^{k=k_0+PHS-1} |y_e(k) - y(k)| \tag{5}$$

At the end of each phase, the best λ is chosen. This is the corresponding value to the lower F. From this λ, new values are generated in a more reduced new R interval

(Fig. 4). The goal is that the identifications performed by the processing units will converge to optimum λ parameters when a given stop criterium will be achieved. So the identification will be of high precision.

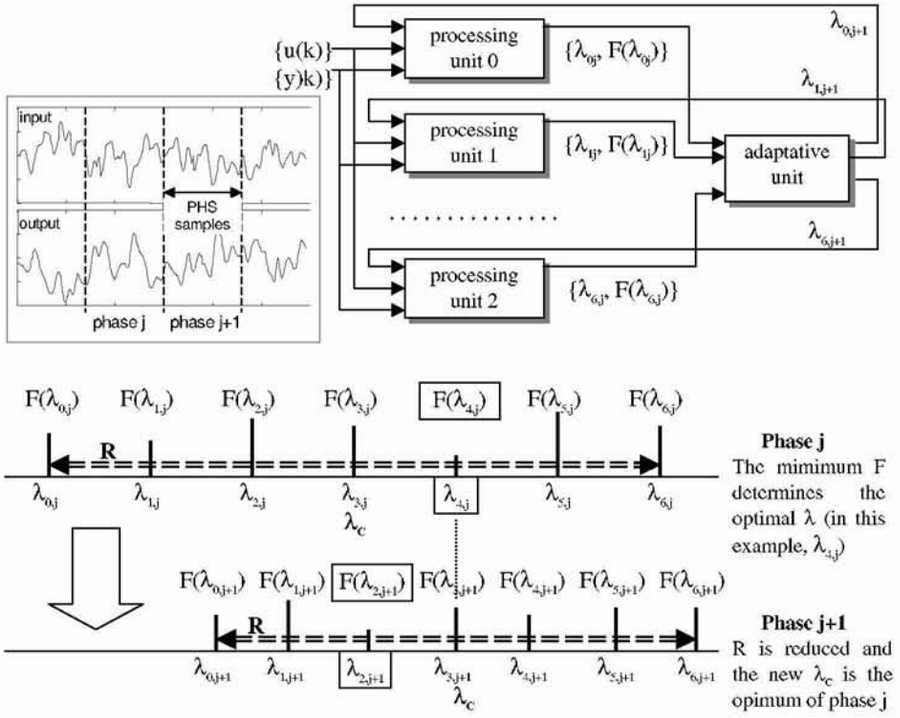


Fig. 4. The SI uses different λ values in each phase performed by each processing unit. All the λ values in the same phase running in the processing units are generated in the R interval from the previous phase optimum λ found, corresponding with the smallest computed F.

2.1. Implementing processing units with neural networks: NNPARLS

Each processing unit of PARLS performs RLS algorithm as it is described in Eq. 3. For other hand, we call NNPARLS to the PARLS version that each process unit is implemented by a neural network performing RLS algorithm. With this purpose we've used the NNSYSID toolbox [7] of Matlab [8]. In this toolbox, the multilayer perceptron network is used [9] because its high capability to model any case (Fig. 5). For this neural net, it is necesssary to fix several parameters: general architecture (number of nodes), stop criterium (the training net is stopped when stop criterium is lower than a determinated value), maximum number of iterations (variable in our experiments), input to hidden layer and hidden to output layer weights, etc.

Our first goal was to determine the best parameters for the neural nets, and the second was to compare both implementations of our parallel adaptative SI methodology to determine which one gets the highest degree of precision. From the first goal, we assigned 5 nodes, stop criterium 0, and weights automatically initialized. The maximum number of iterations was a variable in our experiments in order to study its influence in the NNPARLS performance. By other hand, it was necessary to run many times the same neural net experiment because the results are different each time due to the learning process (Fig. 6).

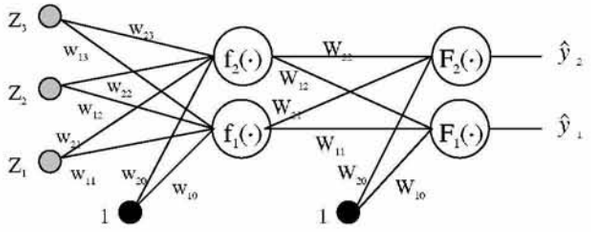


Fig. 5. An example of two layer feedforward multilayer perceptron, used as basis of the neural nets working in NNSYSID toolbox [7].

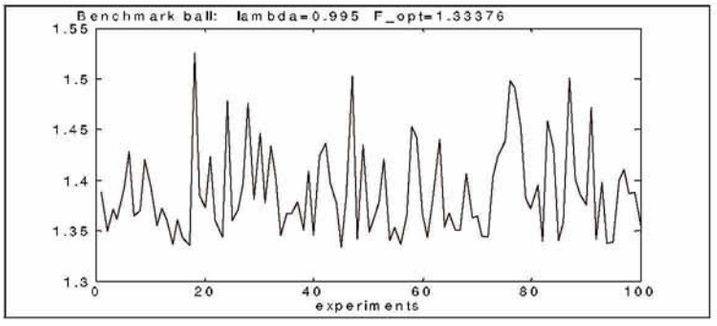


Fig. 6. 100 experiments of *ball* benchmark with the same RLS parameters (λ=0.995) using neural nets. The different result in each experimet is due to the neural net learning process.

2.2. Specifications and parameter tunning

We consider several criteria for PARLS performing. All these criteria have been fully checked and tested [6] in order to get a set of better values for parameters and strategies. For example, we've studied strategies as the optimum λ criterium (the λ value that produces a minimum F), the stop criterium (indicating when a processing unit must stop the work), the model generation criterium (how to consider the initial model in the next phase), the optimum F definition (to consider the optimum F as the lowest in all phases or the lowest computed in the present phase), etc.

PARLS offers a great variability for its parameters. We've carried out many experiments with several benchmarks. From the found results we can conclude there are not common politics for tuning the parameters in such a way that always the best results will be found. But the experience indicates there is a set of values for that the results are good. Then, we establish fixed values for PARLS parameters (Table 2) in order to define an unique algorithm applicable to any system. This has the advantage of an easier PARLS synthesis on a digital architecture (our present effort) and a quickly application for a system without the previous task of tuning parameters.

Parameter	Tuned value
RED	2
PHS	TSN/4 (about 4 phases)

Parameter	Tuned value
PUN	5
Model dimensions	na=3, nb=2, nk=1

Table 2. Main PARLS parameter tuned

3 Experimental results

In Fig. 7 the several experiment results (cost function F) using different methodologies are shown. Four benchmarks with the same model dimensions has been used. Always 5 parallel processing units, 4 adaptative phases, a reduction factor of the interval between phases of value 2 and 0.1 searching initial range centered on $\lambda c=1$ have been considered. (A) shows the found results using NNPARLS for two cases: 20 or 10 iterations for the neural nets training (these results are the arithmetic average of several experiment set). (B) shows the found results using PARLS. (C) shows the results using RLS with 5 equidistant λ values in the initial interval, recording the best case. This has been made because the computational effort for performing 5 RLS processing is similar to the performed by PARLS using 5 parallel processing units. Finally (D) and (E) show the RLS results for two λ values very used in SI.

na=3, nb=2, nk=1, RED=2, PUN=5			initial range		(A) NNPARLS		(B) PARLS	(C) RLS search		(D) λ=.95	(E) λ=.98
bench	tsn	phs	λc	R	F(20)	F(10)	F	λ	F	F	F
ball	1000	250	0.95	0.1	1.32	1.35	1.40	0.975	1.42	1.47	1.42
dryer	1000	250	1	0.1	65.8	67.8	76.7	1	74.6	81.7	77.3
spm1	500	125	1	0.1	137.2	144.7	185.1	1.02	183.9	192.7	186.5
spm2	500	125	1	0.1	129.2	134.6	169.2	1	167.8	176.8	171.3

Fig. 7. Experimental results of several methodologies for 4 benchmarks. All cases consider the same values of the model and parameters, in order to establish valid comparisons.

4 Conclusions

From Fig. 7 we can see NNPARLS always works better than the other cases. This performance is represented in Fig. 8. From these results we can conclude that with

more training of neural net better results, but even in the case of a short training (10 iterations), the found results are always better using neural nets. Also, it is shown the adaptative parallel methodologie without neural nets finds better results than using RLS for several λ values (holding the same computational cost) and a much better result than using RLS for classic λ values (althoug in this case there are more computational cost for PARLS). For these reasons, we can conclude our parallel adaptative methodology optimize the SI, and the neural nets increment the efficiency.

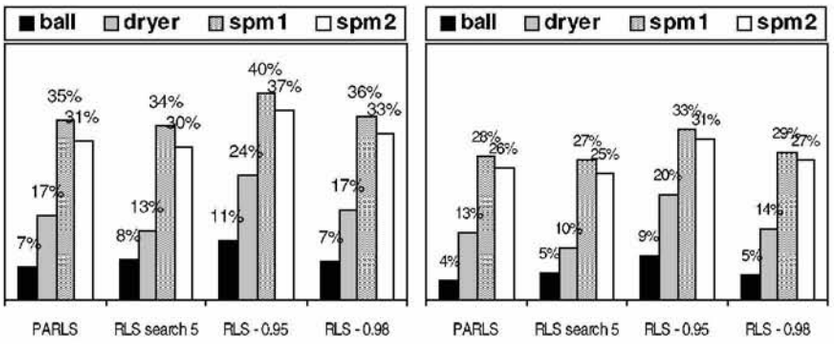


Fig. 8. Comparison of results between NNPARLS and other algorithms (% of worse results): (a) NNPARLS using neural nets with 10 iteration training, (b) 20 iteration training.

NNPARLS and PARLS are not comparible with the computational effort, because the parallel processing units are slower with the neural nets than with the programed RLS algorithm. By that, we are working, at the present, in speeding up these algorithms designing integrated circuits with reconfigurable technology [10].

References

1. Söderström et al., T: "System Identification". Prentice-Hall, London (1989)
2. Ljung, L: System Identification – Theory for the User. Prentice-Hall, London (1999)
3. J.E. Parkum: "Recursive Identification of Time-Varying Systems", Ph.D. thesis, IMM, Technical University of Denmark (1992)
4. Ljung, L: System Identification Toolbox. The Math Works Inc. (1991)
5. "A Database for Identification of Systems". http://www.esat.kuleuven.ac.be /sista / daisy /
6. Gómez, J. et al.: Sintonización de un sistema paralelo para la Identificación de Sistemas. In: Univ. de Extremadura (ed.): I Congreso Español de Algoritmos Evolutivos (2001), 131-138
7. M. Nørgaard: "System Identification and Control with Neural Networks," Ph.D. thesis,Department of Automation, Technical University of Denmark (1996)
8. The Math Works Inc, http://www.mathworks.com
9. Demuth, H, Beale, M: "Neural Network Toolbox". The MathWorks Inc. (1993)
10. Gómez, J. et al.: Diseño de un coprocesador reconfigurable para un algoritmo adaptativo paralelo de Identificación de Sistemas. In: Univ. de Granada (ed.): II Jornadas sobre Computación Reconfigurable y Aplicaciones (2002), 221-226

e e de de g
Ge e A g

d s d z d z z

In o rn ca a a ca ca a a ana a
D ar a n o a a ca ca a n r a a a ana a
[c] D ar a n o c ac on D r nc a n r a a a ana a

A st t a r ro o an r c a roac a on g n c
a gor o o a n n r ca o on or n ca on ro
n r n c na ca g n a o cr o r a on
o o or nar ff r n a q a on o ng
an a ro r a n r ca n gra or an an rror nc on n
ng a g n c a gor x r n w r gn or a o o
I IDS c o on n a

ey r e r y c y e , e e c r h , c e r z e h

n uc i n

d t t t s t d t st d l d l
s t d s t s t s t d l s l ss l p l t t s
s d d t s d t t p l s sts t st
t t w d l p t s z t tw
d l d d t p ll t s p l t s l d tl t
l t ds s ld s d t t t t t d l d t
z t t t
s p p p p s s st pp t t l s l t s
t d t t p l s ffi t t s l t t
d l l pt z t Sp ll t s l t t d l s pp t d
s t l z t t d d t l t s
pl t d t pt z t t
t d s ff t p t l t t d
d ff t l t s s w t t t s s s l 6
dd t ts d l p p t s t p w l t l l t
t 6
t t d s p w l t l l l pt z t
ll w t w d s t sp s l s l t s d
l l s l t s t l t t s d ff t d t st t s
t p s t t s l t s t t t st s t p l
t d t s w t p s t s d t s s s l ss l t ds

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 473-480, 2003.

t t s d l p d p st d t l d t t s l ss s
l p t d t t
p p s z d s ll w s t t st pp
t d s d s d d t l w s ppl d s t t d t t
d l t t d s s t S p d l t ll
s t s z s t l s s d t s t ds t s w

s i i n M

s d t d t st d l s st t d t d s l

$$\frac{\mathbf{x}}{} \qquad \mathbf{x} \qquad \mathbf{x} \qquad \mathbf{x}$$

w **x** s ts st t t t ^{p} ^{p} t
p t s t d l s l t s t s
s ffi t d t s t st d ss s l t s s d
s t d s t s s t s t d l t t s sp t l

y **x** ε

w ε s ll t d p d t d l s w t d s
t t ^{2}
l s t d t w p t s s t t t d st
tw t s l t t d l **x** d t s t **y** s
Sp ll w s t ll w s st t

Θ

w

$$\frac{\mathbf{x} \qquad \mathbf{y}^{\,2}}{^{2}}$$

t s s s t s ε s ss t st t d s w t t
l l d st t s **x** st p t d pp t l
st p t d pp t l s w ll w s t ll w p
p t

Θ

w

$$\frac{\mathbf{x}_{\tau} \qquad \mathbf{y}^{\,2}}{^{2}}$$

d $_{\tau}$ s t pp t **x** t d p t l l

The Me h

t d s d d t s p ss 6

$\mathbf{x}_1$ $\mathbf{x}_n$ $\mathbf{x}_n$

st t t p t $\mathbf{x}$ $\mathbf{x}$ w $_n$ $_n$ $_n$

$_n\mathbf{x}$ $_n$ $_n$ $_n\mathbf{x}$ $_n$ $_n\mathbf{x}$ $_n$ $\mathbf{t}$ $_n\mathbf{x}$

$_n\mathbf{x}$ $_n$ $_n\mathbf{x}$

$_n\mathbf{x}$ $_n$ p $_n\mathbf{x}$

d $_n\mathbf{x}$, $\mathbf{t}$ $_n\mathbf{x}$ d t t d t s t t w t sp t t $\mathbf{x}$ d sp t l l t d t p t $_n\mathbf{x}$ S s pl w p t

The

s st st l t s w s s t d d ls t l t p p l t d d t S l t s t s t p p l t s l s l t s d d ls s s t p l t d s l t s l t p ss s d t d d ls t ss d s p t s t d l t t ss d t t p ss s

t s w s v v v_p p s ts s l l t p t s t p p l t t pp t s p t d d t t s l t d t s ts t ss

ll w t p t s w s d t s l t t l t p ss

S l t t tw d d ls t t p p l t t s l t t s d t d t t st t ss s p t p t s s t t tl tw t t p s t p p l t s z ss tw d d ls v d t t p p l t ffsp v d v st t d s w s l t d tw w t t st t ss t l t t t p p l t **3** d t t d d l v v v v_p t t p p l t w s v v v v_p s st t d pl t l t v w v w s d l s l t d t s t p ss l l s v d d l w t t st t ss s s l t d t l t t t p p l t

t d p t s ll w t st t w p p l t
w t d d ls w t t st t ss s s d
S l t t t s t p s t st d d ls d t
l t t w st s t w p p l t s p t p t s
t t t tw d tw d d ls t st s t
l t t w p p l t t s w t st s s t l st
tw p s t w p p l t w l t w st d s pp s
l ss p t ll ws d d s t tt
d d ls t s t s t t p t s
w ll tt d d d l s s l t d t ss d ts ffsp
l s t s d t s t w ll t t pp t
t t l t s d t
S t s l t d ss p t s s ppl s s
ds tt tt d d d ls t s p l t t
l l d t t p t ll ws d
t s sp d l l s

n i c i n f f A pi ic v u i n in Cu

t d S tw st t s t s t s p ss
t z d l t l ss s d t t ll ss w ff ts
t p pl t s p l
t p t s t p t t l t l t
t S p d s t s p d t t
s p ds t s l t ts w t d p pl d t
z t t d w t d p s ld w l t s t
t s t tw p t t t t ll s p s t s
d t t t
d l S p d l t t t t t t
w ld lp d l d t s t l t t ff t s
t d t p d s z

3 The M e

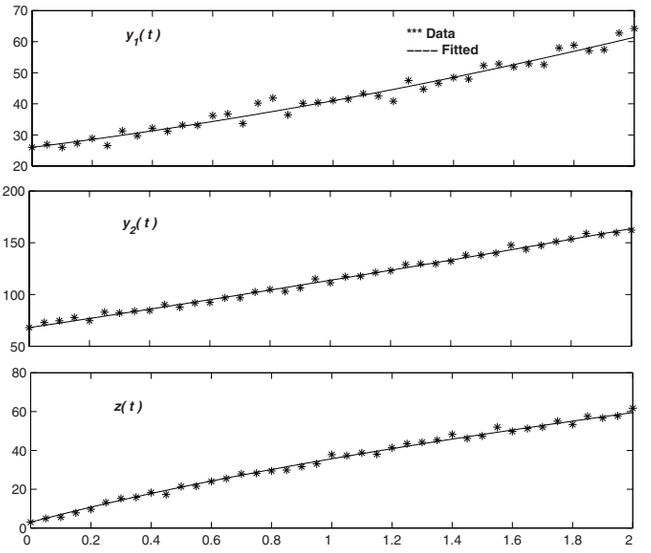
t s s d t ll w d t t t p d d l d l
t d d

$$2 \qquad 2\frac{2}{2}$$

$$2\frac{2}{2}$$

$$\frac{2}{} \qquad 2$$

$$2$$



F I n ca on o I IDS c o on o or 0
r o r a an a a a o an

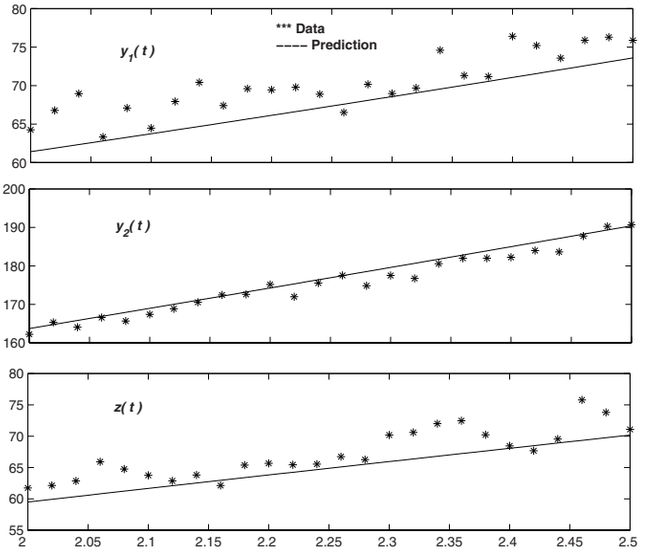
w

w t d p s s t t
w t d p s s t t d t t d t
t t
2 w t d p s s t t d t t d
t w s
p s s w t S t t

2 t w d l s p t s t s
t t d 2 6 6 s t t l st t t
d l t t t s s l
s t d s t s t s **y** w s l t d d t p ss
w t 6 6
2 2 d

3

d t t d st t s t d l s p t s pp p t
s l t t s p t s p p l t s z ss p l t
d t t p l t s d t s l s l t s
w d t t ll w w



F 2 r c on o I IDS c o on o or r o r a an r c a a o an

z t p ss w s p t d t w t d ff t l s t st d l s w 6 d st t d l s s w t l

G ara r

o a on S	00
ro o r ro a 0 00	0
a on ro a	0 0

p ts s w d t t s p p l t s z s l ss t d s t ll w t d p s t t t s sp t t d p p l t s z s t p t t l st s t l p s w t t d t t t dd t ss d t t p l t s t d sp t l ld s ll t s t t d s t d t l w ss d t t p l t s t ll l l w t p l t l s ws t l p t s l s d t st t s st t d t w t d

2 o ara r r r n r a ara r a o
o an ~ r a

			σ					
	0	0	0 0	0	0 00	0 00	0 0	0
	0 0	0	0 0 0	0	0 00	0	0	0

s ws t t t d l t t d t s t
st t s d t d t p d t t d l
pt z t p ss s p t t p t t d t t t d
s p t t l pl t t s d w st p t t t t t
p t t l st t t d p p s d s d tl p p t l t t st
t ts l l t st s l ll t d d l
t p p l t t s pl s ps w l t t
t d s s p t t l st t t d
s t d w s ff l t t s t ppl l t t d t
t d ls w s p t s p dl l d st l
p ss s w t d ls d s l l d l d
t l p w t p t s l sl w w ll w
t s t s t d s s l d t S p d st d d

C nc usi ns

pt z t t d w t s t t d s l t
t s ppl d t t d t t t s t d t st d l
s st s t d s t s t s t d l Sp ll t t d
s ppl d t t d t t S p d d l d l
l s lts s w t t t st t s t d t t d l d
t ffi t st t s t S p d d l d l
t ld ppl d t d t ff l t l l d l d t l
p ss s t l ss t p d t s t d s t s t s t
t t t s t d t s d lt t s s ld
st t d t t tt p d t
p t t l st t t d p p s d t s p p s d tl
p p t l t t st t t lt t pt z t t d
s ld st t d t t

c e e

t s t l s z t st t t d t
t t s s s l l ts d s d
t t p ss s l t s s d z l
p s U s d d d l Sp t l l ll t
t s l t s

f nc s

ra o a o n R non n ar o or a x a ran a
w con ac rac ng I J a o 00
ar J Non n ar ara r a on ca c r Inc
o ng on n on N or o r nar D ff r n a q a on cGraw
N w or
Ja w n S oc a c roc an r ng or ca c r 0
J n J S g ra c x r on o a a oca n ar a on
Sc or S oc a c D ff r n a q a on a r 00
0
J n J ca R ora Ro r g D na c ro r o oca
n ar a on o or n a a ro a o 00

arr ro ara r a on n o n an o r nar D ff r n
a q a on n r ca a roac Doc ora or g na r on n S an
ac a o n r a a a ana 000
c a w c G n c gor Da a S r c r o on rogra
S r ng r r ag r n rg
ro o Ro r g G n c gor a o on or ara r
a on n o n an o r nar D ff r n a q a on D o a
or g na r on n S an ac a o n r a a a ana
00

Input-Output Fuzzy Identification of Nonlinear Multivariable Systems. Application to a Case of AIDS Spread Forecast [1]

J. Ruiz-Gomez, M. J. Lopez-Baldan, and A. Garcia-Cerezo

System Engineering and Automation Department
University of Malaga
E. T. S. Ingenieria Industrial. Campus El Ejido. 29071 Malaga. Spain
ruizg@ctima.uma.es

Abstract. The aim of this work is to show the capabilities of fuzzy modelling applied to a medical problem, the prediction of future cases of AIDS (Acquired Immune Deficiency Syndrome). An automatic knowledge acquisition is achieved from experimental data. This kind of modelling would be useful to practitioners and people not expert in modelling who need a set of fuzzy rules describing the behaviour of some system.
Two modelling techniques have been used in order to obtain the fuzzy models. The first approach is a neurofuzzy modelling technique based on ANFIS. And the second one is a fuzzy method that performs least squares identification and automatic rule generation by minimising an error index.

1 Introduction

Knowledge about a system stated as a formal model is the first step to work with this system. There are different ways to get models, mostly, mathematical methods. But usually these conventional models are difficult of achieving mainly when the knowledge about the system is only a set of input-output data (identification). On identification, the basic dynamic relationship is the linear difference equation and the more common structure is the ARX model and its variants (Output-Error, ARMAX, FIR and Box-Jenkins). State-space models are common representations of dynamical models as well.

An alternative way consists on working with fuzzy models instead of mathematical ones (fuzzy identification). When the system is nonlinear and multivariable, fuzzy models offer some advantages over conventional ones. Rule—based models have been applied to time series forecasting since last eighties [14].

Fuzzy identification is supported by the ability, settled by Kosko [5], Wang [13], and Buckley [4] for approximating any real function, of one or two variables, by means of a set of fuzzy rules.

[1] This work has been supported by the CICYT, Project TAP 99-0926-C04-02.
Thanks to Hector de Arazoza for providing the mathematical model and Liuva M. Pedroso for generating the data.

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 481-488, 2003.

First approaches to fuzzy modelling, based on Sugeno inference method, arose on 1985 by Takagi and Sugeno [12] although Sugeno approach is in continuous development, coming closer to the qualitative reasoning AI viewpoint [11].

Rules, in Takagi-Sugeno approach have the format:

$$\text{IF } x_1 \text{ is } A_1^1, \ldots, \text{and } x_k \text{ is } A_k^1 \text{ THEN } y = p0^1 + p1^1 * x_1 + \ldots + p_k^1 * x_k \qquad (1)$$

And their identification method tries to determine all elements: variables, x_1^i, fuzzy sets, A_k^i and consequent coefficients, p_m^i This identification process produce different models, as several inputs ranges partitions are considered. The range of variable x_1 is divided into two subspaces, and all the other variables are not divided. So it is built the *model 1-1*. In the same manner, occurs with all other variables. For every model the optimum premises and consequent parameters are computed. The procedure stops at a certain model when: 1) the performance index is less than a prefixed value, or 2) the number of generated rules reaches the desired value.

The aim of this work is to show the capabilities of fuzzy and neurofuzzy modelling applied to a medical problem, the prediction of future cases of AIDS. The goal is to estimate both the number of people who will develop Acquired Immune Deficiency Syndrome (AIDS), and the effect on this developing of the number of people that are living with Human Immune Deficiency Virus (HIV) but who have not yet developed AIDS. Estimation is made from the yearly HIV seroprevalence data of the Cuban Partner Notification Program from 1991 to 2000 [1], [4].

AIDS spread models are achieved by using the two modelling approaches, described in the next paragraphs, that accomplish automatic rule generation from experimental input-output data.

The paper layout is: first, an introduction to fuzzy modelling, and a brief overview of the two modelling approaches, a description of the system to be modelled (including the conventional model equations) and, finally, their application to the AIDS problem.

2 Neurofuzzy Modelling

The first modelling method, ANFIS, due to Roger Jang [8] is a hybrid structure based on neural networks and fuzzy logic. It has been chosen due to be one of the most spread out fuzzy modelling approaches (its implementation is included in the Fuzzy Toolbox of MATLAB). Another fuzzy modelling approaches can be used, specially those based on inductive learning [9] or fuzzy clustering [8].

ANFIS (Adaptive-Network-Based Fuzzy Inference System) is an universal approximator that allows a fuzzy inference system to learn by using a backpropagation algorithm based on an input-output data set (learning by an example).

Learning is performed in two stages:

1. the antecedents parameters are kept fixed and the information is propagates to the fourth layer, where the consequent parameters are identified by using the least minimum squares method
2. the consequent parameters are fixed and the error is back-propagated, allowing the antecedent parameters modification by means of a gradient method

This approach is and hybrid learning that combines two optimization methods. The only information specified by the user is the number and type of membership functions and the training sequence.

3 Second Method: Inference Error Learning

Inference error learning stands for a fuzzy modelling method [7], [9] that automatically generate a rule base by introducing new fuzzy sets while learning is necessary.

The input information to the modelling process corresponds to the experimental data obtained from the process. The inference error is computed from real output data and fuzzy output data, and the adjustment is based on least square identification.

The rule base consists of Sugeno type rules in which the consequent parameter stands for the COG of the consequent fuzzy set. The membership functions $A_{nm}(k)$ are triangle sets. The *and* connective corresponds to the algebraic product, and the final output of the fuzzy model inferred from the whole set of implications is given by a modified average defuzzification due to the overlapped condition.

The final model accuracy is expressed in terms of an estimation of inference error, i.e. the difference between the real input/output data and the values generated by the fuzzy model. Particularly, the root mean squared error has been used in the implementation.

The fuzzy model is obtained through an adjustment loop and, after a model is available, it can be further refined with new experimental data.

4 The Raw Data

The data to obtain the fuzzy and neurofuzzy models have been generated from a mathematical model performed by H. Arazoza. [1]:

$$\frac{dX}{dt} = (\lambda - k_1 - \beta - \mu)X + \lambda'(Y1 + Y2) - k_2 - \frac{X(Y1 + Y2)}{X + Y1 + Y2} \tag{2}$$

$$\frac{dY1}{dt} = (-\mu - \beta')Y1 + k_2 \frac{X(Y1 + Y2)}{X + Y1 + Y2} \tag{3}$$

$$\frac{dY2}{dt} = (-\mu - \beta')Y2 + k_1 X \tag{4}$$

$$\frac{dZ}{dt} = \beta X + \beta' (Y1 + Y1) - \mu Z \tag{5}$$

where

- Z is the number of people that has developed acquired immune deficiency syndrome (AIDS)
- (Y1+Y2) is the number of people that are living with human immune deficiency virus (HIV) but who have not yet developed AIDS
- Y1 are the HIV infected that have been detected by means of personal interviews, that is, asking them directly (*contact tracing)*
- Y2 are the HIV infected that have been detected at random
- X is the number of people infected with HIV that has not been detected (this datum is actually unknown and it is not used in this paper)

From these equations L. M. Pedroso has generated a set of 400 numerical data for every variable X, Y1, Y2 and Z, that correspond to a uniform time sequence.

These 400 data are divided in two equal assemblies (rows 1 to 200and rows 200 to 400), generated by using different parameter values for *k1, k2,* β, μ, λ, that represent two different system dynamic behaviours.

Finally, all the data have been contaminated with gaussian noise of median 0 and variance 3, to simulate possible sample errors.

5 Application Results

Two main model sets have been obtained, by using a different data arrange each time:

1. A collection of data arranged as a temporal series, by using only the *Z* variable (people who has developed AIDS), where the antecedents are the three previous values:

$$z(k) = f\{z(k-3), z(k-2), z(k-1)\} \tag{6}$$

2. A collection of data arranged as a first order causal table:

$$y(k) = f\{y(k-1), u(k)\} \tag{7}$$

where output at instant *k* is depends on the input at this moment *u(k)* and the last output, *y(k-1)*. The first order data are arranged by using the variables *Y1, Y2* and *Z,* where the antecedents are the values of the last output *Z(k-1),* the values of *Y1(k)* and *Y2(k)*, and the consequent is the real *Z* output.

For these two groups, two sets of data, that represent different dynamic behaviours, are used to obtain models:

- a set belonging to the first dynamic data array (rows 1 to 200)
- a second one belonging to the second dynamic data array (rows 201 to 400)

Every I/O data set contains 200 rows and 4 columns (antecedents and consequent)

taken at 200 sequenced time instants. This set is divided in two: the first 150 rows as training set and the last 50 ones as checking set.

All the models are evaluated by means of a root squared mean error:

6 First Modelling Approach: Neurofuzzy

By using this first approach, two series of models have been built: models of *Z* as a temporal series and models of *Z* as a causal system from the input variables *Y1* and *Y2*. Though ANFIS allows handling the checking data set in the training process to minimize errors, it is noteworthy that the modelling process has been executed as a pure identification process, without using checking data in the training operation.

6.1 Modelling as a Temporal Series

Three models have been generated arranging data as a temporal series on the *Z* variable, where $z(k) = f\{z(k-3), z(k-2), z(k-1)\}$, with a different training process: 10, 100 and 1000 epochs.

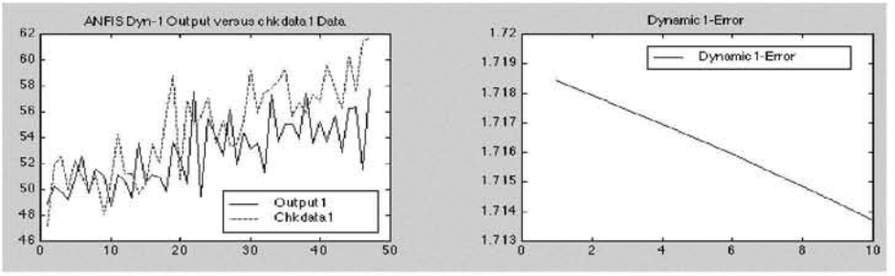


Fig. 1. Dynamics_1 and training epochs=10: Generated data versus checking data, and error

Figures 1 and 2 show that performance improves as training increases, as it is normally expected. These figures correspond only to the to the first block (dynamic type one), the second dynamic type has not beer represented.

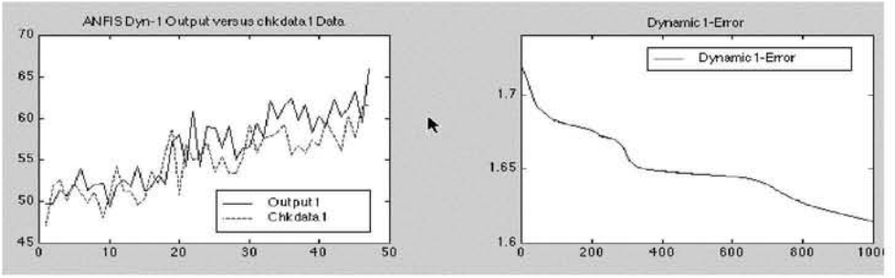


Fig. 2. Dynamics_1 and training epochs=1000: Generated data versus checking, and error

The number of rules has been limited to 8 (two bell shape membership functions for each input variable).

6.2 Modelling as a Causal System

Now, data are arranged as a first order system: $y(k) = f\{y(k-1), u(k)\}$ where the output at the time k depends on the input $u(k)$ and the last output, $y(k-1)$.

Next figures 3 and 4 show the results for the two dynamics (first and second dynamic data sets), for a training process of 10 and 1000 epochs.

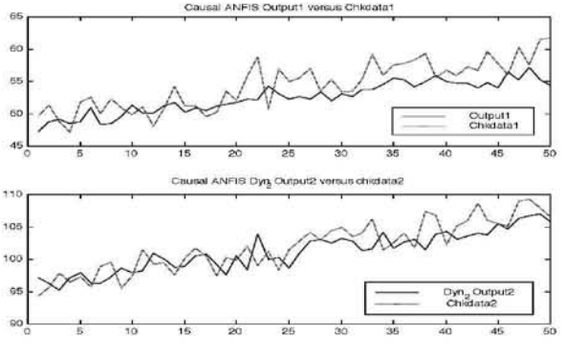


Fig. 3. Output data versus checking data for both Dynamics_1 and Dynamics_2, training epochs=10

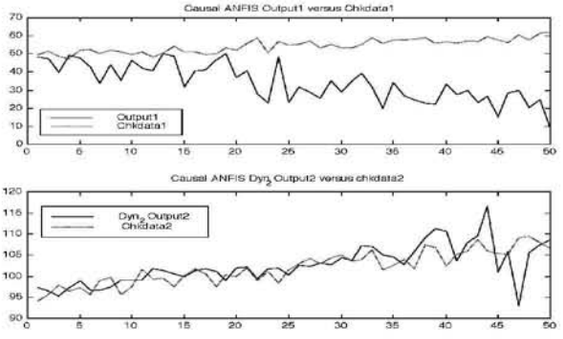


Fig. 4. Output data versus checking data for both Dynamics_1 and Dynamics_2, training epochs=1000

In figure 4, after a longer training (1000 epochs), the errors in the first and second dynamics are unlike. This situation can be explained as a consequence of error

stabilization for the first dynamic system (from 200 epochs on the error is constant), whereas for the second dynamic data set error continues decreasing while training goes on. The points of the first type system become overfitted, on the contrary of the situation in the second type system.

7 Second Modelling Approach

The Inference Error Learning approach has been only applied to model the first dynamic data organized as a temporal series.

With the generated model a good approximation is obtained. Additional work must be done in the future to compare the results with those obtained by means of the first approach.

8 Conclusions

The main result on fuzzy modelling applied to the spread of AIDS is that a good prediction of future AIDS cases is possible, based both in the previous values of registered events (temporal series prediction) or in the case of first order causal models (based on the values of *z(k-1), y1(k), y2(k)*).

This good approximation is accomplished by using both modelling approaches (Neurofuzzy and Inference Error Learning ones).

Considering equal training (10 epochs), better results are got for causal models, due possibly to a stronger relation between variables *Z(k-1), Y1(k)* and *Y2(k)* with the output *Z(k)* than the temporal series relation between *Z(k-3), Z(k-2), Z(k-1)* and the real value of *Z(k)*. The strong of these relations, as input variables sorting according to their relative importance, can be analyzed by using inductive learning techniques based on ID3 and EG2 algorithms [7], [9]. This is a topic for further work.

In causal models, results goes better for the second dynamic data as the numbers of epochs increases, while it is opposite for the first dynamic data. This is probably due to an over-fitting for the first dynamic data.

Nevertheless, a good approximation between original and modelled data is achieved for both dynamics, in spite of working with models of a low number of rules (eight rules). However, this allows improving the training time (0.68 seconds for 10 epochs and 53.19 s for 1000 epochs, in causal models and by using a Macintosh iMac with MATLAB 5.0).

Though the presented approaches have been applied to a medical case, other applications have been accomplished to I/O identification based on numerical data. Some of these are: the modelling of an autonomous mobile robot dynamics (RAM-2, built in the University of Malaga) based on velocity and curvature experimental data measured after a short travel [7], and the modelling of a ternary batch distillation column [9]. The design of fuzzy controllers based on human actions (cloning) is a common application in control system problems. And due to the short training time, the first approach can be applied to design adaptive controllers that generate new rules on line.

References

1. Arazoza, H., Loumes, R., Hoang, T., Interian, Y.: Modelling the HIV epidemicunder contact tracing. The Cuban case. Journal of Theoretical Medicine, *2* (2000) 267.274
2. Buckley, J. and Hayashi, Y.: Fuzzy input-output controllers are universal approximators, Fuzzy Sets and Systems 58 (1993) 273-278.
3. Garc a-Cerezo, A.J., L pez-Bald n, M.J, Mandow, A.: An Efficient Least Squares Fuzzy Modelling Method for Dynamic Systems, Procced. CESA 96 .IMACS Multiconf. Symp. Modelling, Analysis and Simulation. Lille. (1996) 885-890.
4. Hsieh, Y. H, de Arazoza, H, Lee S. M , Chen, CW : Estimating the number of Cubans infected sexually by human immunodeficiency virus using contact tracing data. International Journal of Epidemiology Vol 31, Iss 3 (2002) 679-683
5. Kosko, B.: Fuzzy systems as universal approximators, Proc. Int. Conf. on Fuzzy Systems, San Diego (1992)
6. Ljung, L.: System Identification Toolbox. For use with MATLAB. The Mathworks, Natick (Mass.) (1995)
7. Lopez-Baldan, M. J., Ruiz-Gomez J., Fernandez, R and Garcia-Cerezo, A., Input-Output Fuzzy Modelling Applied to a Mobile Robot. In Mastorakis, N. (ed.): Computational Intelligence and Applications. World Scientific and Engineering Society Press (1999) 283-288.
8. Roger Jang, J.-S., Sun, C.-T: Neuro-Fuzzy Modeling and Control. Proceed. IEEE, 83, No. 3, (1995) 378-406.
9. Ruiz-Gomez, J., Lopez-Baldan, M.J., Garcia-Cerezo, A.: Fuzzy Modelling of a Ternary Batch Distillation Column. Int. Journal of Computer Research 11, No 3 (2002) 347-355
10. Sugeno, M., Kang, G.T.: Fuzzy Modelling and Control of Multilayer Incinerator, Fuzzy Sets and Systems, 18 (1986) 329-346
11. Sugeno, M., Yasukawa, T.: A Fuzzy-Logic-Based Approach to Qualitative Modeling. IEEE Transactions on Fuzzy Systems 1, No 1 (1993) 7-31.
12. Takagi, T., Sugeno, M.,: Fuzzy Identification of Systems and Its Applicattion to Modeling and Control. IEEE Transactions on Systems, Man and Cybernetics, 15. (1985) 116-132.
13. Wang, Li-Xin: Fuzzy Systems are Universal Approximators. Proceed. Int. Conf. on Fuzzy Systems, San Diego (1992) 1163-1170.
14. Zardecki, A.: Rule-Based Forecasting. In: Pedricz, W. (ed.): Fuzzy Modelling. Paradigms and Practice. Kluwer. Boston Dordrecht London (1996) 375-391

Re e g g d
B ye e

st ll l S z ² p l s t z s² d
st ll

D ar n o a a c an o a ona Sc nc
n r o an a r a 00 San an r S a n
tie u i e
D ar n o o r Sc nc n r o or ña
0 or ña S a n
e i i u e i u e
D ar n o on n o an c an c an S r c ra or
n r o Grana a 0 Grana a S a n
ti ug e

A st t a r r n o o or r co r ng ng
a a ng nc ona an a an n wor In ca o a a
o ng a a on can con r ng a a a ar a an arn
og r w o ara r n n a on roc I
on con rar ng a a arg on can arn nc ona
or n ra n wor ro co a a an o arn ng
a a on ca a a na o xa o a ca on o
ra o o og ar r n ow ow ng a a
r co r g n ra a n r o ng a a r ca ncr a
ng an a n ona rror a r a a ow a r c co ar on w
ca o a ng a a In a on a an n wor a roac
o g r r an nc ona n wor

n uc i n

p l ss pl t d t s p t ss
pl s st t p d t t w t s l l t s s d t
d t st d t s l st t s ll s l t d t p d t w l
s st t t w t d t s w ds s w ds t t s t
t t s t pp t s st t st t s l t t
d d t s s s t s st t s s d t p t l
t t s ppl t t s d pp s t
l t ss d t t d t t st t s d t
l t s st t s
st pp p t w t dl ss pl t d t w ll d p d
p w d t p ts ss ttl d d t
t p s ss d t s s

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 489-496, 2003.

pl t l d ss s s t t d w t t
ss l s s t t s s w t pl t d t d st s l
s s w t pl t d t
d ss s s t d w t t ss l s d
s s w t ss d t p d t l t l s
l ss d t t p d t l t l s

s w d ll t t ds dl ss d t S t
p p l t ds

s w s d t d l t s s w t ss d t d t
l s s
w s d t d l t t l t s s p t
st t st s s d p t l l p w s d t t t l
t pl t d t s s
s st t t pl ss d t t l s t d
t s s pl t t
l l d pp Us ll l l d t t t
l l d s d s ffi t st t st s d t s
lt pl p t t t s t l w d t l s t ll ps d t
d t s s l l d
6 ss t ds Us ss t t d t
d t pl t t t p d t ss d t
t d p t t pl t ss l s t s ss t d w t
t st s l s
p t t z t pp w ll w l t
s s d t st t t ss d t p t d l s l l t d t
p t t st p t t t s d t z t l l d t
z t st p t l s tt d

t d ttl d p t s t ds d l d t t
s w s p w s d s st t t t ds t ss
l l d t ds lt pl p t t
t s p p w p p s l d s tw s pp s t s l
t ss d t p l l d s tw s t s l
s d s s t t w d s l d w d l pt d t d l
t d l w t p t l p l s sts t t p s tw s l d
s s l t ls t d l t s tw t p t d t tp t l s
ss d t st t s l t s s d t st t t s d t

A y si n w App c

s tw s sp ll d t d t ss d t p l
s t t t pd t t p l t s l s w
s t t t l s s w

e *B o k od l B o k od l o l*
B o k D D d d l
{ *n n of o d o l ob b l d (D*
o fo bl d of of od D T
d o d o ob b l d

n

d t s s tw s

t d s t s s l s pl d s l d d t s
d t l p l t s
d p d l t s l s s l d t
p
d t l p l t s s t l s t l s
t l pp s t l s s l t d

ss t ds p d t t l t s s
tw s ll w t l t p d t t t t w l d t l d st
t t sp d l dd t t sp ll d s d t
s l t s p l d t t l t s t
t s s w t t l d d t s t s l t ss p l
s tw d ls s l d s t t s d
l s
t st s s t s l s tw s ts t t l
s s l t ll w t

The re *o d o l of u d bu o d b o*
of do bl ul u d bu o -
o d o b

d *Z*
Z Z ZZ

d o d o of Z d
ZZ o d o of d Z o
of d T D of ul u
o Z d o Z b

Z Z ZZ Z

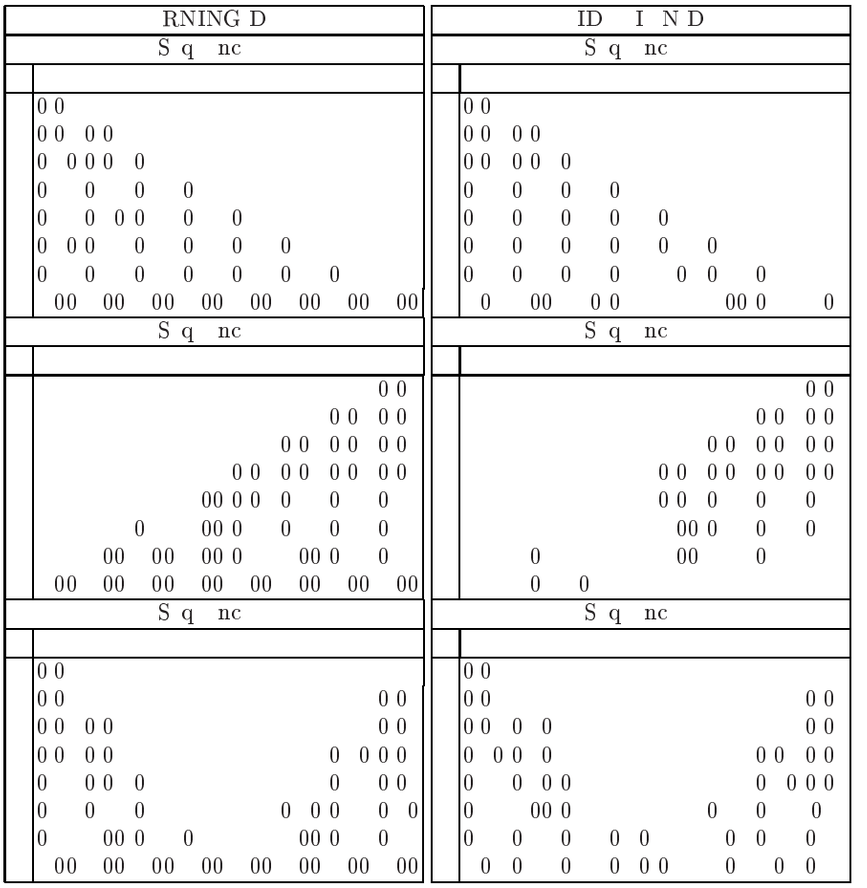
Z Z ZZ Z

t t t l s d ll w d t t d t l d st
t s ll ws t l p t st t s t t l
st t s w t t t t p d t
t s p p w s l s tw s t ll st t t ss d t
p l t t s tw d ls ld s d

$$\epsilon$$
$$\epsilon_2$$
$$\epsilon$$
$$\epsilon_4$$

A unc i n w App c

T e St d d z d t s d ss t d w t p d t d
l s t d ff t l s s ss d t
s s s d d



w 2 p 2 s p t l
t s { 2 s t s t p t s d ϵ s d l
w t d st d d d t
ss t t s t d t s st t s t d t l s

{ 2 p 2

S p d t st t t p t s s sts s l t
pt z t p l s st ll t l

n
z p 6

w { 2 d t t t ll w

p p

2

p p

p p

w s t d t l s ss t s t
p d t s s t w p s t tw t ds ss
d t
t s t ss d t l s s s ll p t t ss
d t l s s l s t t z t p ss d t ss d t
st t d s t s t l ss t t t s t pl t d t s s ffi tl
l d t s s ffi tl st l t tw t p t d t tp t
l s
p s l ss t t t s t p t d tp t ss d t

{ s ss { s ss

t t l 6 pl d t p l

n

z 2 p 2

w w t ss d t l s d t pt z t p ss
st t s t s l s

R k t s t st t t t t ss d t t p s
t t tw t p t d tp t l s st st

R k l t t s s w t t tp t
l s s t s pl s st ts t d t

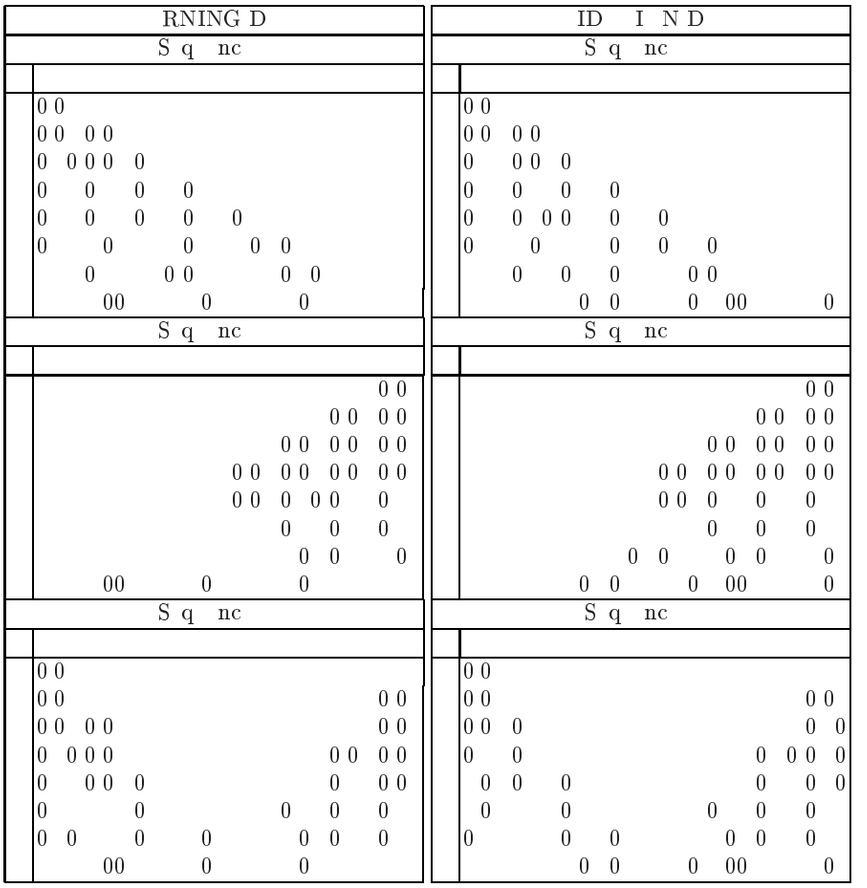
lt t s sts l t t 2 p s
pl t d t pl s l t p l 6 d t s t s lt
t t l t ss d t s s l t p l

z 2 p 2

s p ss st p t d ll w t s ss d t z
t p ss ll t ss d t l s t st t d ss
t d d

l s d t s d t s t s pl d t s ss
d t s s l s ws t p t s t d
p s l s d s ws t t t s tw pp
s tt st t s t t t l tw pp

T e St d d z d t s d ss t d w t p d t d l s t d ff t l s s ss d t s s s d d



C nc usi ns

l s s t t t d t s p p

t l tw s s l t s l t ss d t p l s tw s sp ll d t t s l t ss d t p l s t t d t t t d t l d st t s s t l s t s t d s t s tw s ts d t d w t t s t ss d w d t ss d t d d t s w t t ss d t p s t t d p d t l s s t s d d t sl w t p t d

Ac n w n s

t s d t d t t Sp s st S d l
t d t p t S p t t t
U s t p t l s pp t t s p t

f nc s

a o o o J G rr an r n a *nc na*
ca n w r ca c r o on Dor r c on on
a o nc ona N wor *a c ng*
a o an J G rr Non n ar S r o ng an r c on
ng nc ona N wor x rac ng In or a on a ao *y c*
o
a o o o J G rr an R r n a nc ona N wor
N w N ra N wor a o o og *an n a c*
ng n ng o 0 0 000
a o J G rr o o an a o n ax o or arn ng
nc ona N wor *a c ng* 000
a o o o R Go N r n an a G n ra ra wor or
nc ona N wor 0 000
a o J G rr o o an a o So arn ng o
n nc ona N wor *an n a c ng n ng*
000
a o J G rr S a an acr So ca on o
nc ona N wor n S a c an ng n r ng *c n c* o No
0 00
R J R n D *S a ca ana y ng a a* N w
or Jo n an Son
0 ar J *bab c a n ng n n g n Sy a*
b n nc organ a ann San a o
Ro ng a a conc a r w or a c o og
nn yc gy 0

f peed e f y
e e

ll ll tt † l zz †

Sc o a S r or San nna o o San nna a ra
a R na o agg o 0 on ra I I
D ar n o n rg ca S S cco n r a r n
a S ar a r n I

A st t a r cr an co ar o a ca on o n ro
N o a o a ra n ro a r
n o oc o wo ax n an w /ra a on con on
r n N a roac o r or r gn cr
a gor n r o co a ona r n an o o ac
a o a gn ca r or anc ro n on ra ng r
ca a o arn ng ro ro g a a

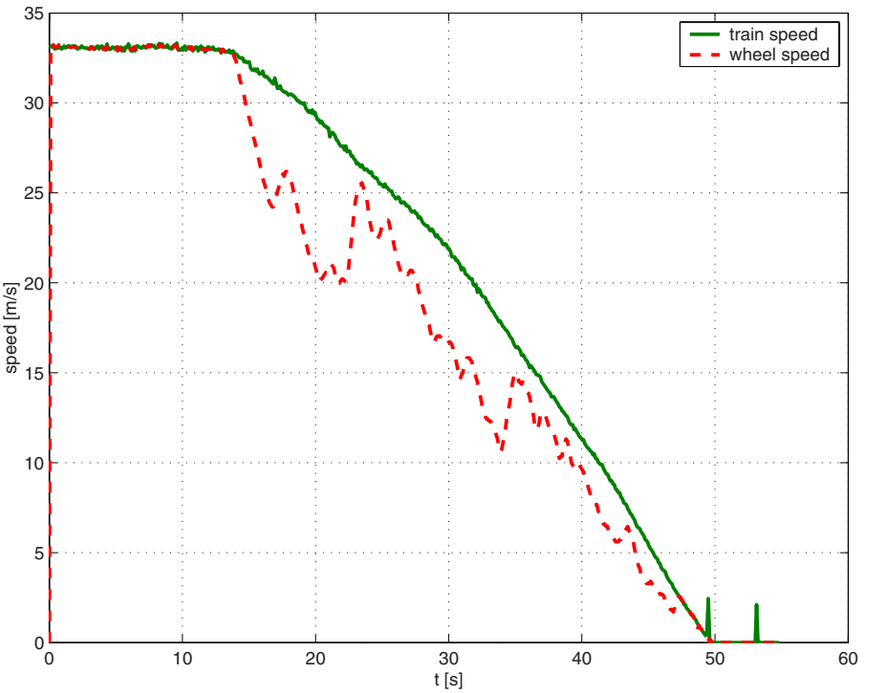
n uc i n

t t t t s st s ll p s d
ou d ub t t pd t s t t p s t d l d t t
pl t s s t s l t d t s st d l s s l t d
l t l d *o bo d ub* w st t s t t l t sp d
d p s t d s s st p d s t d w t t t
d st d p s t s t sp d l t t t t
t t p t t l d t l d t d s s st s
l t s t d st $_n$t t ll ws pl t sp d l t
t t t t p t d t d st t t t t p ts
pl t t t l t sp d t p t s d t t sp d
s t d st t w t t t sp d s d ppl t
d l t t d ff tw $_n$ d t d st
t t t p ts s s ll t d l t d s s st
t t s t l l st t t t l t l t
s t s l t l t t s d l s s s s t t t
sp d ts t t t t p ts
t s st d S d l p d t l Sp d t
U s t l pp st t t l t s t t sp d
d d st t t t p t l t d p ss t
s ts p d d tw t l d s p s t d tw d p d t
l s s p p p s d ff t d p t sp d st t s
p p s s d s ll ws t p l t t sp d st t s
p s t d S S d s s pp s d t d s

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 497-503, 2003.

s st w l lt t l st t s d s d S t
s st s p s p d S w l S 6 d s ss s t s lts
d t d l s t w

sc ip i n



F ra n a n n an w a n n a ra ng r or
w a ng c n gra a on con on

d t t s s d s s s l t d l s t
t s d d d tw tw s s t t p s t
s ts w t t w ls d t t ls t t sp d v
s t s s t t v v R w s t l sp d t
t w l d R s t w l d s t t w t
d s d t s d d d w s t tl w t t
s l t d l s d t s l t l
t s p s t t p ll d t s l ld d
l s t d t s s s sl d s t t t l l t st

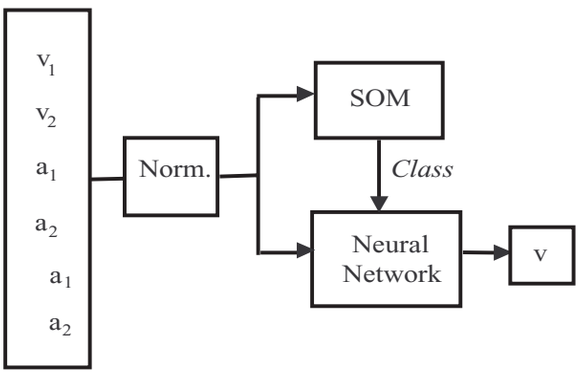
u y v u i n f in sp

A n iv n u i s

IS	S G N
N r o n	
or ac n	
r nc on	Ga
ara r or ac	
No	
n ar ara r	0
Non n ar ara r	
o a ara r	
ra n ng a a a r	
r	
ra n ng oc	0

arac r c o N IS or a on

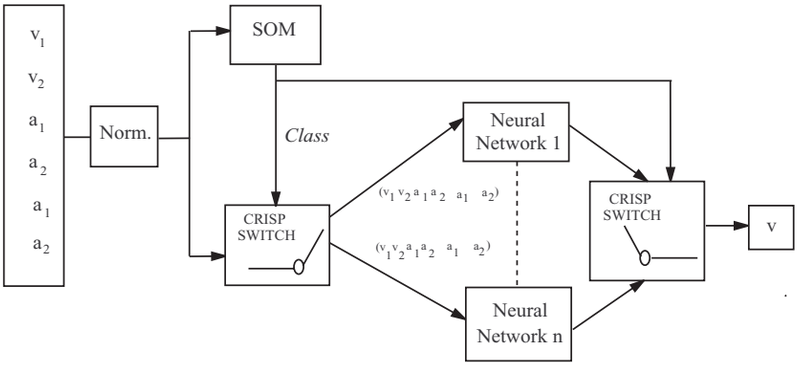
d t p t p tw l l st t s t st d t l d d t s tt pts w s t l t d st s t d ff t s t t s t t s ld t d ff t d p t p tt s st d d s d t s w l t w l l t s d l t s s l d t l t l s t d t l l *b fo* t st t s l l t w t s d t t l sp l t S l s p S s s d d t l ss t s d s l p tt s st tt pt t s lt t l s s t s d tl p t s l p t l tw t t w t t s d s l p t p tt s d p t d



F 2 Sc o r rarc ca n ra gn or ra n a on

s d tt pt l ss l sp d st t s d s d t s lts t l ss t s s d dd s d s l

p tt t t st t tw s d t t l ss t l s t
s s w



F Sc o con rarc ca n ra a or a r or a ff r n
a a on accor ng o ff r n r o c a ca on o n
a rn

u ic su s iscussi n n c p is n

p t d ff t sp d st t s t d t t
l s d s t s S S sp d st t

$$f \quad — \quad — \quad \hat{v} \quad v \quad ^2$$

w s t t p t s t pt z d s t
t d t d s t st d d d t t t sp d
s ws t st s lts t d w t t d ff t t sp d st
t s l t s t p t l t p t ls
t t d d s d t l t s t t l s t t st
p st t l ss t d s t l d s t l t t
l t st t st t t ls t t s l t t st
p t l s l t st t t l t t st t s st
s l l s st w s l l t s d s d t s l t
t t t p l d t d s t l s st t t ts t p l
t st
l l s ws t t t d s t sp l t w s pl
d t s w l t t d d s t S w s s s l
s t p d t S w s t % w t
sp t t t s d t pt s d t l sp l t

A o t t p	s t s	t	v t o t s	
r n a a w G		00		0 0
N IS		00	0	0 0
n		0		0 0 0
n 0		0		0 0
S n		0		0 0 0
S n				0 0

2 o ar on a ong a gor n r o r q r or gn ra n ng on n a a roc ng an o r or anc

t d s t s l tw st t d ff t tw s w t s s t dd l t st d t p d tt pts t lt l p pt l d t t st s t l st t d t pt l p w s d w t dd s ll t tw s w t d w t t *b u d* t d tw w t 6 p ts tp s t l d pt s d s s t sp l t w t st t p t % d % sp t l

t tw l l st t s t t S t p l t p tt l ss w s st t w d t d ff t t p l s t d t s d st s tw s s t st s w t d s p t l stl t S s d s d d ff t l ss s t t w ds s l tt pts p d d t s t t p l d t d s t ll w d w d tw ls t s s s t s pl tw l d st t tp s t t s pt l s l t d t d ff t t s s p t st t d p t d t pt l s l t s sts S w t l ss s d w t dd s w l t st t d p t d l ss s s ffi t d t ll w s dd s s ll tw s d s s st t s ll t d t t d t sp d l t t s d l l st t t t s ls t st p st t s t S S s d d % w t sp t t t s l w t 6 p ts d % w t sp t t t pt s d sp l t

C nc usi ns n fu u w

p l t sp d st t t s t t l t tw l s ll t d s d t s s d w t t s pl t d ff t pp s l t zz pp d t s w tt p s w t sp t t t sp l t d s d d t d t s t t l p t p s l l s st s p t l t d

s l t s s w d t ffi p d tt st t d
t p w t d s l t p l s sts t pl t t
l w d t t l sp l t l t s d ff t
t ds sp d st t d t t d s t d t s d t t
s d t s l d pt t p l t s st s
d s d s lts d st t d t t lt s st s
s p l t l d t t d s t s l t
p t l p l st t d w t d p p s t p l
ts l s t p t t l t t t t t t t l
p s l p d s d t l t s t
d s S w t s d t t ld t l s s pl t t
p t s t s w l w t t s d d p l
t w s s d t p ss l t t t t l
s s st s t l st t s s t s st s d t d t
t t s s t t t w s st ll t d d t
s st s d l l ll t s t t s s s t t t l
s sp t t l p s l t t t d s lts

f nc s

N n na o ra n ra ng acc rac an a ro c on anc n a o
a c ra n ro c on *c n n n*
gn an ac an a n n a ay an anc a an
Sy 1
r c an a G onacc a D o n o a ra ng
o or r on *c 1 ng a ay a c*
1 o n G r an No r 00
S rog o a gor o r ca co o a oc a an an a r no
o a o rcor o R/D S S r n a a S n a cno og a
r a Ro a c o r 000
og c n con ro og c con ro r ar an
an Sy an an yb n c o 0 No 0 0
J S ng R Jang N IS a N wor a In r nc S
an Sy an an yb n c o No
o a ann cc o a a o ar on o ra ona an
n ra or ra n a on n r *c* 11^t *Sy n*
c a a S r g g r 00
ang n n a N wor *an ac n n a*
o No No r
o on n organ ng a *c ng* o No
0 0
D arq ar n a gor or a q ar a on o non n ar ara r
S na a a c No

Neuro-Fuzzy Techniques for Image Tracking·

José M. Molina*, Jesús García*, Javier de Diego+, Javier I. Portillo+

* Universidad Carlos III de Madrid, Departamento de Informática
Avenida de la Universidad, 30. Leganés 28911. Spain
+ Universidad Politécnica de Madrid. E.T.S.I. Telecomunicación
Ciudad Universitaria s/n.Madrid 28040. Spain
Email: jgherrer@inf.uc3m.es

Abstract: This work presents the application of neuro-fuzzy techniques to develop a rule-based fuzzy system as an efficient and robust approach for object tracking based on sequences of video images. The fuzzy system is a new approach for data association problems in video image sequences, which uses JPDA formulation adapted to cope with video data peculiarities. The neuro-fuzzy learning algorithm uses examples extracted from a surveillance system of airport surface, including situations with very closely spaced objects (aircraft and surface vehicles moving on apron). The learned fuzzy system ponders update decisions both for the trajectories and shapes estimated for targets with the set of image regions (blobs) extracted from each frame. The inputs of the neuro-fuzzy system are several numeric heuristics, describing the quality of gated groups of blobs and predicted tracks, and outputs are confidence levels used in the update process. Rules are aimed to generate the most appropriate decisions under different conditions, emulating the reasoned decisions taken by an expert, and have been learned by the neuro-fuzzy system. The system performance with real image sequences of representative ground operations is shown at the end.

1. Introduction

The Advanced Surface Movement, Guidance and Control Systems (A-SMGCS [1]) receives and processes data from different types of sensors. This work is centered on the tracking system operating on sequences of video images provided by cameras deployed on the airport surface. The general architecture and main blocks integrated were described in [2]. Each processor calculates target trajectories (local tracks) in the projected camera plane, performing two steps. First, moving targets are detected against their local background to generate detected pixels, connected then to form compact image regions referred to as blobs. Blobs are defined with their spatial borders, generally a rectangular box, centroid location and area. Then, the tracker must distinguish all targets in the scene and track their motion, applying association and filtering processes.

· This work has been funded by Spanish CICYT, TIC2002-04491-C02-01/02

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 504-511, 2003.

The data association logic is one of the basic aspects determining the system capability to cope with dense, multi-target scenarios [3]. Its design must take into account the characteristics and quality of processed data. In this case, data are the blobs, resulting from the detection subsystem applied on image sequences of airport surface scenes.

In this work, a fuzzy system for association is developed applying Nauck/Kruse neuro-fuzzy to obtain the rules. In section two, the specific problems in this application are analyzed. The third section presents the fuzzy approach to evaluate the confidence levels used to update estimators describing targets shapes and motion parameters. The learning method and the description of the learned rules are shown in section fourth. Finally, system output in several scenarios is presented in fifth section, indicating the response for complex situations and the performance of the rules learned with real image sequences of representative ground operations.

2. Problem Definition

A first problem to be considered is the imperfect image segmentation, due to image irregularities, shadows occlusions, etc., resulting in multiple blobs potentially generated for each target. This splitting effect may appear randomly, with irregular surface vehicles or shadows, or systematically when some obstacle or target partially occludes other interest targets moving behind. So, blobs must be re-connected, deciding which ones are to be grouped and associated to each target. On the other hand, closely spaced targets lead to overlapping images, appearing some targets partial or totally occluded by other targets or obstacles, so that some blobs can be the result of incorrect segmentations and really represent to several targets.

Therefore, each available frame to process presents a set of blob-to-track multi-assignment problems to be solved, where several (or none) blobs may be assigned to the same track and simultaneously several tracks could overlap and share common blobs. An all-neighbors approach taking soft decisions, similar to JPDA [4], seems adequate for this problem, since all blobs potentially gated with each track are used to update it, requiring moderate computation load. Usual formulation of JPDA is not directly applicable to the problem with extended targets, since residuals are computed assuming simplified models considering the centroids, predicted positions and certain error distributions. It is more adequate to extend the association, using an explicit representation of target shape and dimensions, together with motion estimation, to select the set of updating blobs for each track. The evolution of shape, accordingly to the information obtained from the images sequence, should be updated jointly with motion parameters. There are not detailed models or analytical expressions to design this process, similar to JPDA, but an analysis of continuity performance with different strategies, depending on numeric heuristics describing the situations, provided robust rules to take appropriate association decisions [5].

The proposed approach presented in this work is based on the knowledge of target shapes and dimensions to extract and correlate the appropriate set of blobs with each track. The dynamic evolution of these estimations, targets shapes and trajectories, in accordance with the information extracted from images, is the key aspect to guarantee

good performance figures regarding continuity. A fuzzy system [6,7] is proposed to evaluate the confidence given to the information contained both in the gated blobs and predicted tracks, based on a set of numeric heuristics describing the characteristics of each situation. When we can not derive the labels' membership functions of linguistic variables (input/output) or fuzzy rules from the expert knowledge, an automatic learning procedure could be applied. Although many proposals have been developed based on neural networks [8] or on genetic algorithms [9], the learning techniques based on neural networks show mathematically consistence and have been applied profusely in many applications [10]. The techniques based on neural networks are named neuro-fuzzy systems and they are usually represented as a multilayer feedforward neural network [11]. Two approaches of neuro-fuzzy systems exist. The first one uses differentiable operators in the fuzzy system to apply gradient descent procedures, these systems generate fuzzy systems that are not easy to interpret. The ANFIS model by Jang [12] implements a Sugeno-like fuzzy system [13] in a network structure, and applies a mixture of backpropagation and least-mean-square procedure to train the system. The GARIC model [14] uses a special "soft minimum" function which is differentiable. The second one uses max-min operators and the learning procedure is heuristic, these systems are easy to interpret as the neuro-fuzzy systems developed by Nauck and Kruse (NEFCLAS [15] and NEFCON [16]). The fuzzy system for association uses Mamdami implication [7] because the fuzzy system interpolates a generic function (the association function) that has not analytical expression. We have applied the Nauck/Kruse neuro-fuzzy approach because they use directly this type of implication and the method is developed for this type of fuzzy systems.

3. Fuzzy System for Multiple Cameras Tracking

The first steps are the detection and extraction of blobs in a frame. Then, the tracking system tries to correlate blobs with estimated tracks. The association (blobs to tracks) will be performed by means of a correlation mask, predicted to the frame time instant from the last update. This mask conforms to the estimated shape of the target, and will be updated also with the spatial information contained in the blobs. Therefore, track-state vectors with position and cinematic estimates (2D location and velocity referred to the camera plane) are complemented with attributes defining a spatial representation of target extension and shape. This predicted target contour is used to gate blobs extracted in next frame. For the sake of simplicity, a rectangular box has been used to represent the target, as indicated in figure 1. Around the predicted position, $(\hat{x}_p, \hat{y}_p)$, a rectangular box is defined, (x_{min}, x_{max}, y_{min}, y_{max}), with the estimated target dimensions $(\hat{I}_H, \hat{I}_V)$. Then, an outer gate, computed with parameters Δ_H, Δ_V, is used to finally gate the potential blobs updating the track estimates.

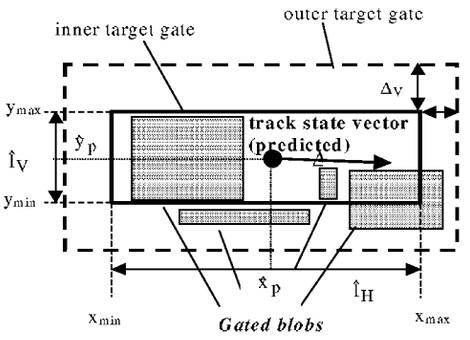


Figure 1. Target segmentation with estimated box

This outer gate allows the system track dynamic variations in target shape along the sequence, for targets not perfectly matching to predictions due to variations in projected shape (changes of orientation, distance, etc.), or maneuvers. Besides, it avoids the initialization of tracks around existing ones, potential source of instabilities. The shape must be dynamically updated with the information contained in blobs, accordingly with the set of rules. The changes must be smooth, avoiding instabilities in complex scenarios.

As any multi-target tracking system, one of the key points of the whole tracking system will be the data association logic, which is the focus of interest in this work. An analysis of conventional systems performance under different conditions has been used to propose some heuristics assessing the characteristics of different situations and the most adequate association decision to take. This set of heuristics will be combined with a fuzzy system to produce a final "correlation level" of each blob to update each possible track, with a weighting formulation similar to JPDA [4]. They were detailed in [7]: (a) Overlapping heuristic, this component can be seen as a "soft gating", computed as the fraction of blob area contained within track predicted region; (b) Group density and distance to track, this heuristic evaluates the ratio between areas of detected regions and non-detected areas (holes) in the finally reconnected pseudo-blob; (c) Conflict with other tracks heuristic, this component evaluates the likelihood of blob being in conflict with other tracks; (d) Proximity to image borders heuristic, this number evaluates if the blob is close to any of the four image borders.

Heuristics defined above will be the input to unknown functions computing the confidence levels both for blobs and predicted tracks. A rules system based on fuzzy logic has been developed in order to approximate these functions [7]. The rules have been obtained by analysis of conventional tracking systems performance under different conditions, depending on values of heuristics. Fuzzy reasoning techniques may be adopted to reproduce these behavior under the conditions specified in the automatically learned rules, and besides generate the proper output for all intermediate cases.

4. Rule Extraction using Neuro-Fuzzy Techniques

The data-analysis process to derive the decision rules from available data is summarized here. The well-known NefClass algorithm [15] was applied to data sets with the heuristics characterizing input images and tracker output in sample scenarios. The scenarios were selected considering representative situations such as image splitting in segmentation, occlusions and overlapping, in order to get a robust system able to attain acceptable behavior in the general case. The three possible outputs considered for blob confidence regarding track parameters update were {discard, low, high}. So, for each detected blob located around the target bounds (predicted by the tracking system), the three possible decisions were: accept the blob and update the estimated track parameters (high), if the blob information is reliable and only referred to the represented target; discard the blob (discard), if it clearly comes from a different source; or partially update the track (low), when the blob has information about target but it is corrupted by effects such as occlusion or overlapping. The three selected scenarios had the following characteristics:

- Scenario 1: Three aircraft moving in parallel taxiways, in an area covered by a camera with low depression angle, so their images get overlapped when they cross. The confidence level of blobs in such situations must be lowered to avoid degradation of estimated shape and kinematics of targets.
- Scenario 2: Three vehicles were moving on a road, approaching until their images get overlapped, while at the same time one of them performs a deceleration maneuver. Figure 2 shows some sample frames, where a van (white) and a vehicle are moving from left to right, while other vehicle is moving from right to left. Images from the three vehicles get overlapped when the first vehicle moving from left to right stops in front of the aircraft, white van overtakes it, and at the same time the other vehicle crosses behind. Besides, figure 3 indicates the detected output and tracker output in four important instants where the tracks were updated accordingly to the rules in [7].
- Scenario 3: this scenario contains multiple blob reconnections. An aircraft is moving behind stopped vehicles and aircraft. These vehicles occlude it and other vehicles move in close parallel roads. Multiple blobs representing different parts of aircraft and its shadow appear. These blobs must be grouped to update the aircraft track, avoiding splitting effects. Besides, images from other vehicles must be kept separated guaranteeing track continuity for all targets.

So, for each one of the frames available in the scenarios, the blobs and tracks were used to compute the heuristics, while the label describing the confidence category of blob was manually assigned from direct observation.

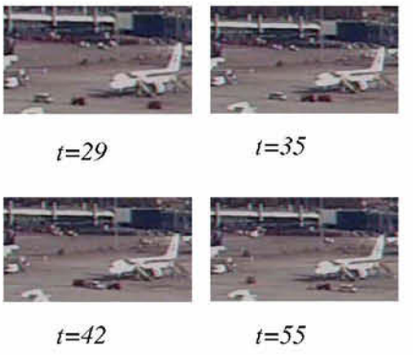


Figure 2. Scenario 2 vehicles with multiple overlaps

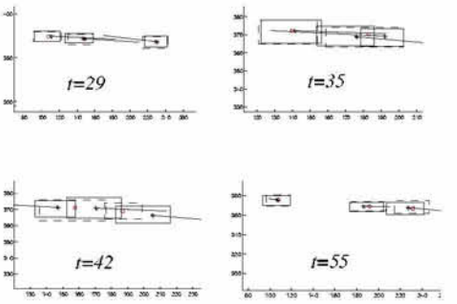


Figure 3. Tracker output for scenario 2

The set of rules generated with all available data (around 250 frames, with 700 instances composed by 6 attributes and output), and using bell-shaped sets appears in figure 4. As it can be seen, the most important attributes to classify blobs are conflict degree and overlapping with track.

- if overlapT is large and density is medium and proximity is large and conflict is small and border is small and overlapI is large then HIGH
- if overlapT is large and density is large and proximity is large and conflict is small and border is small and overlapI is large then HIGH
- if overlapT is large and density is medium and proximity is large and conflict is small and border is medium and overlapI is large then HIGH
- if overlapT is large and density is large and proximity is large and conflict is small and border is medium and overlapI is large then HIGH
- if overlapT is small and density is large and proximity is large and conflict is large and border is small and overlapI is large then DISCARD
- if overlapT is small and density is medium and proximity is medium and conflict is medium and border is small and overlapI is large then DISCARD
- if overlapT is small and density is medium and proximity is medium and conflict is large and border is medium and overlapI is large then DISCARD
- if overlapT is small and density is medium and proximity is medium and conflict is large and border is small and overlapI is large then DISCARD

Figure 4. Rules obtained

5. Results

Finally, in order to analyze the quality of the classification scheme and its capability to predict the tracker decisions, the total set of data was split in subsets for validation. The three scenarios were divided into ten groups, depending on the characteristics of each sequence (segments without conflicts, segments with occlusions, with bad segmentations, etc.).

The training and evaluation process was performed with different scenarios to obtain the rate of instances correctly predicted, as depicted in table 1. Two type of

fuzzy sets were selected to represent the concepts: triangular and bell-shaped, whose performances are indicated in the top and bottom, respectively, of each cell. From the results, the learning capability is better when bell-shaped functions are applied. The main diagonal is blank since the test was always performed over data different from training. Final column has the mean performance with each training set applied to the rest of available data. The worst results were obtained when data from simple segments without problems were used for training. For instance, intervals (22-43) and (53-88) from scenario 1 contained only separated targets, while segments (44-52) and (61-67) conflict situations. The rules generated with the second type obtained better results than those generated with the first one.

Table 1. Accuracy of rules generated with different subsets of data

Test / Training	S1 (22-43)	S1 (44-52)	S1 (53-88)	S1 (61-67)	S2 (72-137)	S2 (115-120)	S3 (40-63)	S3 (64-85)	S3 (86-111)	S3 (112-136)	All All
S1(22-43)	- -	36,7 43,3	89,5 94,7	33,3 53,3	91,6 99,0	28,0 44,0	88,2 94,1	48,4 60,5	46,2 65,1	73,8 100,0	61,9 76,0
S1(44-52)	100,0 93,5	- -	94,7 81,6	63,3 63,3	92,6 74,7	56,0 64,0	88,2 79,4	68,6 67,7	63,9 60,3	73,8 70,2	76,2 70,2
S1(53-88)	100,0 100,0	83,3 86,7	- -	73,3 86,7	92,6 100,0	68,0 84,0	91,2 94,1	76,6 96,8	69,2 94,7	76,2 100,0	78,8 95,7
S1(61-67)	100,0 96,8	83,3 90,0	94,7 96,1	- -	94,7 84,2	56,0 80,0	91,2 82,4	74,2 67,7	70,4 74,6	77,4 96,4	80,7 81,2
S2(72-137)	100,0 100,0	43,3 43,3	89,5 94,7	33,3 53,3	- -	28,0 44,0	88,2 94,1	50,0 60,5	50,3 65,1	94,1 100,0	63,9 73,7
S2(115-120)	100,0 100,0	83,3 86,7	94,7 96,1	63,3 76,7	94,7 92,6	- -	91,2 91,2	85,8 95,1	73,4 89,4	75,0 73,8	83,4 89,6
S3(40-63)	100,0 100,0	43,3 43,3	94,7 94,7	43,3 53,3	93,7 99,0	28,0 44,0	- -	53,2 60,5	53,3 65,1	77,4 100,0	67,2 76,2
S3(64-85)	100,0 100,0	83,3 86,7	100,0 100,0	73,3 86,7	92,6 96,8	60,0 84,0	91,2 94,1	- -	75,7 91,7	77,4 85,7	83,8 92,5
S3(86-111)	100,0 100,0	83,3 86,7	94,7 100,0	63,3 86,7	94,7 99,0	72,0 76,0	91,2 94,1	84,7 96,0	- -	75,0 100,0	85,8 95,9
S3(112-136)	100,0 100,0	43,3 43,3	94,7 94,7	43,3 53,3	93,7 99,0	28,0 44,0	94,1 94,1	55,7 60,5	56,8 65,1	- -	68,7 73,9
Repres.1	100,0 100,0	- -	- -	63,3 76,7	94,7 100,0	68,0 84,0	91,2 97,1	- -	75,2 94,1	75,0 97,6	80,8 94,9
Repres.2	- -	100,0 86,7	94,7 100,0	- -	94,7 96,8	72,0 84,0	91,2 97,1	87,1 97,6	- -	76,9 83,3	88,4 93,8
All	- -	- -	- -	- -	- -	- -	- -	- -	- -	- -	**84,1** **93,3**
N.Instances	31	30	76	30	95	25	34	124	169	85	699

The specialization effect can be noticed since the rules obtained for each scenario obtain better results than those generated from other ones (although with different data sets than training), but worse in the rest. The best results were obtained with a representative set containing a sample of conflictive situations in scenarios 1 and 2 (row labeled as representative 2), with some overlaps, occlusions and splits. Finally, the table entry All represents the result obtained with a random sampling for training and validation complementary sets, applying cross validation, very similar to the best result. In fact, both sets drove rule sets very similar to the one in figure 4, generated with all available data.

6. Conclusions

Neuro-fuzzy learning techniques have been successfully applied to solve the core problem of data association for video tracking under complex, high-density conditions. Specific domain knowledge is represented as a set of rules to adapt association decisions as a function of several heuristics inferred from experimentation.

7. References

[1] U.S. Department of Transportation. FAA. The Future Airport Surface Movement Safety, Guidance and Control Systems: A vision for Transition into the 21st Century. Washington. November 1993.

[2] J. A. Besada, J. Portillo, J. García, J. M. Molina. Image-Based Automatic Surveillance for Airport Surface FUSION 2001. Montreal, Canada. August 2001.

[3] S. Blackman, R. Popoli. Design and Analysis of Modern Tracking Systems. Artech House. 1999

[4] Z. Ding, H.Leung, L. Hong. Decoupling joint probabilistic data association algorithm for multiple target tracking. IEE Proceedings- Radar, Sonar and Navigation, Vol 146, N°5. pp 251-254. October 1999

[5] J. García, J. A. Besada, J. M. Molina, J. Portillo. Fuzzy data association for image-based tracking in dense scenarios. IEEE International Conference on Fuzzy Systems. Honolulu, Hawaii. May 2002

[6] L. A. Zadeh, Outline of a New Approach to the Analysis of Complex Systems and Decision Processes, Trans. on SMC, Vol. 3, N° 1. January 1973

[7] J. M. Mendel. Fuzzy Logic Systems for Engineering: A Tutorial. Proceedings of the IEEE. Vol. 83. N°3,. pp 345-377. March 1995

[8] JJ Buckley, Y. Hayashi Neural Networks for Fuzzy Systems. Fuzzy Sets and Systems, 71, 265-276. 1995.

[9] V. Matellán, C. Fernández, J. M. Molina, Genetic Learning of Fuzzy Reactive Controllers. Robotics and Autonomous Systems, vol 25, n° 1-2, pp 33-41, 1998.

[10] S. K. Halgamuge, M. Glesner, Neural Networks in designing Fuzzy Systems for Real World Applications. Fuzzy Sets and Systems, 65, 1-12. Wiley, 1994.

[11] N. Tschichold-Gürman, Generation and Improvement of Fuzzy Classifier with Incremental Learning using Fuzzy Rulenet. Proc. ACM Sym on Applied Computing, Nashville, pp 466-470, 1995.

[12] JSR Jang, ANFIS: Adaptive-network-based Fuzzy Inference Systems. IEEE Transactions on System, Man and Cybernetics 23, 665-685, 1993.

[13] M. Sugeno, An Introductory Survey of Fuzzy Control. Information Science 36, 59-83, 1985.

[14] HR Berenji, P. Khedkar, Learning and Tuning Fuzzy Logic Controllers through Reinforcements. IEEE Transactions on Neural Networks 3, 724-740, 1992.

[15] D. Nauck, R. Kruse, NEFCLASS- a neuro-fuzzy approach for the Clasification of Data. Proc. ACM Sym on Applied Computing, Nashville, pp 461-465, 1995.

[16] D. Nauck, R. Kruse, R. Stellmach New Learning Algorithms for the Nuero-Fuzzy Environment NEFCON-1. Proc. Nuero-Fuzzy Systems'95, armstadt, 357-364, 1995.

A Comparative Study of Fuzzy Classifiers on Breast Cancer Data

Ravi. Jain[1] and Ajith. Abraham[2]

[1]School of Information Technology, James Cook University (Cairns Campus), Smithfield, Australia 4878. ravi.jain@jcu.edu.au
[2]Computer Science Department, Oklahoma State University (Tulsa Campus), 700 N Greenwood Avenue, Tulsa, Oklahoma OK 74106, USA. aa@cs.okstate.edu

Abstract. In this paper, we examine and compare the performance of four fuzzy rule generation methods on Wisconsin breast cancer data [2]. These methods were reported by Ishibuchi [1] et al. For the diagnosis of breast cancer, the determination of the presence of *benign/malignant* breast tumors represents a very complex problem (even for an experienced cytologist). The goal of this paper is to compare and contrast fuzzy rule generation methods on breast cancer data that involve no time-consuming tuning procedures. Since The performance of each approach for test patterns (i.e., the generalization of ability of each approach) is evaluated by cross validation techniques on breast cancer data sets.

1. Introduction

Breast cancer is the most common cancer in women in many countries. Most breast cancers are detected as a lump/mass on the breast, by self-examination/mammography, or by both [3]. Screening mammography is the best tool available for detecting cancerous lesions before clinical symptoms appear. Surgery - either a biopsy or a lumpectomy have been the most common method to remove them. Fine needle aspiration (FNA) of breast masses is a cost-effective, non-traumatic, and mostly non-invasive diagnostic test that obtains information needed to evaluate malignancy. Recently, a new minimally invasive technique, which uses super-cooled nitrogen to freeze and shrink a non-cancerous tumor and destroy the blood vessels feeding the growth of the tumour has been developed [4] in USA. Artificial Intelligence techniques (such as neural networks, fuzzy logic, genetic programming and combination of these methods) have attracted a lot of attention in the area of medical diagnosis. These techniques are successfully applied to a wide variety of decision-making problems. Several methods have been proposed for generating fuzzy if-then rules for pattern classification problems. Neural network architectures and learning mechanisms were used for tuning fuzzy if-then rules. Genetics-based machine learning frameworks were used for generating and selecting emphasized the direct generation of fuzzy if-then rules from training patterns. While many sophisticated approaches have been proposed, very simple fuzzy if-then rules were also used as fuzzy classifiers. The main aim of this paper is to examine the performance of some direct rule generation methods that involve no time-consuming tuning procedures on *breast cancer data*. The first one generates fuzzy if-then rules using the mean and the standard deviation of attribute values. The mean and the

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 512-519, 2003.

standard deviation were used as parameters of membership functions when they compared fuzzy classifiers with neural network classifiers. The second approach generates fuzzy if-then rules using the histogram of attributes values. In the first two approaches, a single fuzzy if-then rule is generated for each class. The third approach generates fuzzy if-then rules with certainty each attribute into homogeneous fuzzy sets. In the fourth approach, only overlapping areas are partitioned. This approach is a modified version of the third approach.

2. Rule Generation Procedure

In this section, we explain each of four approaches examined in this paper. The performance of each approach is evaluated by computer simulations on breast cancer data sets.

Assuming that we have an n-dimensional c-class pattern classification problem whose pattern space is an n-dimensional unit cube $[0,1]^n$ and m patterns $x_p = (x_{p1},\ldots,x_{pn})$, $p = 1,2,\ldots,m$, are given for generating fuzzy if-then rules where $x_p \in [0,1]$ for p =1,2,..., m, i =1,2,...,n. In computer simulations, all attribute values are normalized into the unit interval [0,1].

3. Rule Generation based on the Mean and the Standard Deviation of Attribute Values

In this approach, a single fuzzy if-then rule is generated for each class. The fuzzy if-then rule for the k^{th} class can be written as

If x_1 is A_1^k and ... and x_n is A_1^k then Class k, (1)

where A_i^k is an antecedent fuzzy set for the i^{th} attribute. The membership function of A_i^k is specified as of

$$A_i^k(x_i) = \exp\left(-\frac{\left(x_i - \mu_i^k\right)^2}{2\left(\sigma_i^k\right)^2}\right) \quad (2)$$

where μ_i^k is the mean of the i^{th} attribute values x_{pi} of Class k patterns, and σ_i^k is the standard deviation. Fuzzy if-then rules for the two-dimensional two class pattern classification problem are written as follows:

IF x_3 is A_3^1 and x_4 is A_4^1 THEN Class 2 (3)

IF x_3 is A_3^2 and x_4 is A_4^2 THEN Class 3 (4)

The membership function of each antecedent fuzzy set is specified by the mean and the standard deviation of attribute values. For a new pattern $x_p = (x_{p3}, x_{p4})$, the winner rule is determined as follows:

$$A_3^*(x_{p3}).A_2^*(x_{p4}) = \max\left\{A_1^k(x_{p3}).A_2^k(x_{p4})\,\middle|\,k = 1,2\right\} \quad (5)$$

For each attribute 20 membership functions $f_h()$, h=1,2,...,20 were used. The fuzzy partition used only for calculating the histogram.

4. Rule Generation based on the Histogram of Attribute Values

In this method the use of histogram an antecedent membership function and each attribute is partitioned into several fuzzy sets. We used 20 membership functions $f_h(.)$, h=1,2,...,20 for each attribute in computer simulations.

The smoothed histogram $m_i^k(x_i)$ of Class k patterns for the i^{th} attribute is calculated using the 20 membership functions $f_h(.)$ as follows:

$$m_i^k(x_i) = \frac{1}{m^k} \sum_{x_p \in Class\ k} f_h\left(x_{pi}\right) \quad (6)$$

$$for\ \beta_{h-1} \le x_i \le \beta_h,\ h=1,2,...,20$$

where m_k is the number of Class k patterns, $\left[\beta_{h-1}, \beta_h\right]$ is the h^{th} crisp interval corresponding to the 0.5-level set of the membership function $f_h(.)$:

$$\beta_1 = 0,\ \beta_{20} = 1, S \quad (7)$$

$$\beta_h = \frac{1}{20-1}\left(h - \frac{1}{2}\right) \text{ for h=1,2,...,19} \quad (8)$$

The smoothed histogram in (6) is normalized so that its maximum value is 1. As in the first approach based on the mean and the standard deviation, a single fuzzy if-then rule in (2) is generated for each class in the second approach.

5. Rule Generation of based on Simple Fuzzy Grid

In the first two approaches, a single fuzzy if-then rule was generated for each class using the information about training patterns. On the contrary, many fuzzy if-then rules are generated in the third approach by partitioning each attribute into homogeneous fuzzy sets.

One disadvantage of this approach is that the number of possible fuzzy if-then rules exponentially increases with the dimensionality of the pattern space. For coping with this difficulty, some GA-based rule selection approaches have been proposed to find a compact rule set [7,10,12]. The number of fuzzy if-then rules can be also decreased by feature selection [14].

Because the specification of each membership function does not depend on any information about training patterns, this approach uses fuzzy if-then rules with certainty grades. The local information about training patterns in the corresponding fuzzy subspace is used for determining the consequent class and the grade of certainty.

In this approach, fuzzy if-then rules of the following type are used:

IF x_1 is A_{j1} and ... and x_n is A_{jn}

THEN

Class C_j, with CF = CF_j, j =1,2,…, N (9)

where j indexes the number of rules, N is the total number of rules, A_{ji} is the antecedent fuzzy set of the i-th rule for the i-th attribute, C_j; is the consequent class, and CFj is the grade of certainty. The consequent class and the grade of certainty of each rule are determined by the following simple heuristic procedure [13]:

Step 1: Calculate the compatibility of each training pattern x_p =(x_{p1},x_{p2},…,x_{pn}) with the j-th fuzzy if-then rule by the following product operation:

$$\pi_j\left(x_p\right) = A_{j1}\left(x_{p1}\right)\times\ldots\times A_{jn}\left(x_{pn}\right),\ p = 1, 2, ..., m. \quad (10)$$

Step 2: For each class, calculate the sum of the compatibility grades of the training patterns with the j-th fuzzy if-then rule R_j:

$$\beta_{class\ k}\left(R_j\right) = \sum_{x_p \in class\ k}^{n} \pi\left(x_p\right),\ \text{k=1,2,...,c} \quad (11)$$

where $\beta_{class\ k}\left(R_j\right)$ the sum of the compatibility grades of the training patterns in Class k with the j-th fuzzy if-then rule R_j.

Step 3: Find Class A_j^* that has the maximum value $\beta_{class\ k}\left(R_j\right)$:

$$\beta_{class\ k_j^*} = Max\{\beta_{class\ 1}(R_j), ..., \beta_{class\ c}(R_j)\} \quad (12)$$

If two or more classes take the maximum value or no training pattern compatible with the j-th fuzzy if-then rule (i. e., if $\beta_{Class\ k}(R_j)=0$ for k =1,2,..., c), the consequent class C_i can not be determined uniquely. In this case, let C_i be ϕ. If a single class takes the maximum value, the consequent class C_j *is* determined by (7).

Step 4: If the consequent class C_i is 0, let the grade of certainty CF_j be $CF_j = 0$. Otherwise the grade of certainty CF_j *is* determined as follows:

$$CF_j = \frac{(\beta_{class\ k_j^*} - \overline{\beta})}{\sum_{k=1}^{c} \beta_{class\ k}(R_j)} \quad (13)$$

$$CF_j = \frac{(\beta_{class\ k_j^*} - \overline{\beta})}{\sum_{k=1}^{c} \beta_{class\ k}(R_j)} \quad (14)$$

6. Rule Generation based on Fuzzy Partition of Overlapping Areas

In the third approach, the shape of each membership function was specified without utilizing any information about training patterns. A simple modification of the third approach is to partition only overlapping areas.

This approach generates fuzzy if-then rules in the same manner as the simple fuzzy grid approach except for the specification of each membership function. Because this approach utilizes the information about training patterns for specifying each membership function as in the first and second approaches, the performance of generated fuzzy if then rules is good even when we do not use the certainty grade of each rule in the classification phase. In this approach, the effect of introducing the certainty grade to each rule is not large if compared with the third approach. In computer simulations of the next section, we used fuzzy if-then rules with certainty grades in this approach as in the third approach.

7. Simulation Results

The Wisconsin *breast cancer* dataset was obtained from repository of machine learning database University of California, Irvine. This data set has 32 attributes (30 real-valued input features) and 569 instances of which 357 benign and 212 malignant class. There are several studies based on this database. Bennet and Mangasarian [19] used linear programming techniques, obtaining a 99.6% classification rate on 487 cases (the reduced database available at the time). However, diagnostic decisions are essentially black boxes, with no explanation as to how they were attained.

In the first approach, a single fuzzy if-then rule was generated for each class using the mean and the standard deviation of attribute values. A single fuzzy if-then rule for each class was not sufficient for the breast cancer data. In the second approach, a single fuzzy if-then rule was generated for each class using the histogram of attribute values. The third approach generated fuzzy if-then rules by homogeneously partitioning each attribute. Thus a pattern space was partitioned into a simple fuzzy grid. The information about attribute values was not used for specifying the membership function of each antecedent fuzzy set. The local information of training patterns was utilized when the consequent class and the certainty grade were specified. The 10-fold cross-validation procedure was iterated 5 times using different partitions of data sets into ten subsets. Simulation results are summarized in Table 1. In this table, the 85.96% classification rate by the fuzzy classifier systems is the best result among simulation results. From this table, we can see that the performance of the fuzzy rule-based classification systems except for the histogram approach is comparable to the best result by the fuzzy classifier systems.

Table 1. Simulation Results

Methods	With CF	Single winner rule (without CF)
Mean & Deviation	85.94%	85.94
Histogram	62.74%	84.36
Simple Grid	62.39	62.39
Modified Grid	62.57%	85.96

8. Conclusion

In this paper, we examined the performance of four fuzzy rule generation methods that could generate fuzzy if-then rules directly from training patterns.

Grid-based approaches, had high generalization ability but the number of fuzzy if-then rules is exponentially increased with the dimensionality of the pattern space. Thus a large number of fuzzy if-then rules are usually generated for real-world pattern classification problems. This leads to several drawbacks: overfitting to training patterns, large memory storage requirement, and slow inference speed. The last approach was a modified version of the simple fuzzy grid approach. Only overlapping areas of different classes were partitioned into fuzzy subspaces.

On the contrary, the number of fuzzy if-then rules in the first two approaches is the same as the number of classes. As shown in this paper, a single fuzzy if-then rule

for each class is not always sufficient for real-world pattern classification problems. While each approach is very simple and has the above-mentioned drawbacks, generated fuzzy rule-based systems have high classification ability. The performance of fuzzy rule based systems can be improved by feature selection and rule selection.

References

1. Ishibuchi, H., and Nakashima, T., A Study on Generating Classification Rules Using Histogram Edited by Jain L.C., and Jain, R.K., KES'98, pp. 132-140
2. Merz J., and Murphy, P.M., UCI repository of machine learning databases. http://www.ics.uci.edu/˜learn/MLRepository.html, (1996)
3. DeSilva, C.J.S. et al. Artificial Neural networks and Breast Cancer Prognosis The Australian Computer Journal, 26, pp. 78-81,(1994)
4. The Weekend Australian, Health Section, p 7. July, 13-14, (2002)
5. Cios, K.J., et.al. Using Fuzzy Sets to Diagnose Coronary Artery Stenois, IEEE Computer, pp. 57-63, (1991)
6. Ishibuchi, H. et.al Pattern and Feature Selection By Genetic Algorithms in Nearest Neighbour Classification unpublished report.
7. Ishibuchi, H. et.al, A fuzzy classifier system that generates fuzzy if-then rules for pattern classification problems, Proc. Int. Conf. Evolutionary Computat. Perth, Australia, vol. 2, pp. 759-764, (1995)
8. Pal, S. K. and Mitra, S. Multilayer perceptron, fuzzy sets, and classification, IEEE Trans. on Neural Networks, 3, no. 5, pp. 683-697, (1992)
9. Nauck, D., and Kruse, R., A neuro-fuzzy method to learn fuzzy classification rules from data, Fuzzy Sets and Systems, vol. 89, pp. 277-288, (1997)
10. Ishibuchi, H., Nozaki, K.,Yamamoto, N., and Tanaka, H., Selecting fuzzy if-then rules for classification problems using genetic algorithms, IEEE Trans. on Fuzzy Systems, vol. 3, no. 3, pp. 260-270, (1995)
11. H. Ishibuchi, H., T. Nakashima, T., and Murata, T., A fuzzy classifier system that generates fuzzy if then rules for pattern classification problems, Proc. 2nd IEEE International Conference on Evolutionary Computation, pp.759-764, Perth, Australia, (1995)
12. Yuan,Y., and Zhuang,H., A genetic algorithm for generating fuzzy classification rules, Fuzzy Sets and Systems, vol. 84, no. 1, pp. 1-19, (1996)
13. Ishibuchi, H., Nozaki, K., and Tanaka, H., Distributed representation of fuzzy rules and its application to pattern classification, Fuzzy Sets and Systems, vol. 52, no. 1, pp. 21-32, (1992)

14. H. Ishibuchi, H., Nakashima,T., and Morisawa, T.,Simple fuzzy rule-based classification systems performed well on commonly used real-world data sets, Proc. of North American Fuzzy Information Processing Society Meeting, Syracuse, pp 21-24, (1997)

15. Weiss, S.,M., and Kulikowski, C.,K., Computer Systems That Learn, Morgan Kaufmann Publishers, San Mateo, CA, (1991)

16. Goldberg, D.E., *Genetic algorithms*. Addison-Wesley Publishing Company, (1989)

17. Goldberg, D.E., Genetic and evolutionary algorithms come of age. Communications of the ACM, vol. 37, No. 3, pp. 113-119, (1994)

18. Zadeh, L.A., Fuzzy Logic IEEE Computer, pp. 83-93 (1988)

19. Bennett, K.,P., and Mangasarian, O.L., Neural network training via linear programming. In P. M. Pardalos, editor, Advances in Optimization and Parallel Computing, pages 56-57. Elsevier Science, 1992

A e f e e f ge

st d t d s s

D o Ing n r a In or a ca an In o Ing n r a onoc n o
n r a ono a a r 0 a r S a n

A st t o g na ra ag ar a r a o a
ag r c co a on o r oc n no o n
o r an cc o o ag roc ng o o ar
c ar rac a co r on n ca a o ow ar a o
ca r a a ar o na ra ag a c In wor w a
ow ow a concr roc r a a rac a ag co r on
can o r q r c an an ar anc nor a oc
a c a o n an ag n ro a r an a
an ag r r n a on an c r r a on w o r w
ag n or a on a r

n uc i n

s t ll p ss l s t l l s s t l
t t d t l s t t s t s d d t l s s
d st t t t t s t d p t l t
t t t ll ws t p ss t s l s st t
t l t s t d st t d l t t
p ts t t st l t t s l s st w l t p ts l ss
l t t t l sss pt t t t p p s t s l s st s
t t t t s p ts w t l t p t t st s
w t s t s t t t t t s st s t l l t
s s l p ss t ls d t
p ss s st s s s st ffi t p ss
t l pp t t t t s t s t d t st
t p t l st t st s t t l s s ld p ss s d l ff t s
d t t t d t 6 w t s t s t s t t
d t t l st t st s w ll t d ffi lt t st t
s t t s t p l ssl ss p ss t d l
s p ss t t ts p p l s t ts
d d t p ss p l s w pl s t t ds t s
[4] l s t p t s l l l s s t t t
s d tl t p ss l s s l d t t st d s l s pl d
d ls w s st t st s s t p t d t s pl s
t s l ll d l ls l l s d ts s ll

[⋆] ar a or o S a n I I 0 an 0

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 520-527, 2003.

t s t st s p t d d d l s lt
s l t t s l t t l ls d l
d t s d d p w st t st s p s t st l l s pp
t t l st t st s w t s s t l st pl t t l
st t st s t p ss t s p t l l t
pl s t l p ss ffi t t l p ss
ss t ll d p ds t s ss l pt s d st t st s
t s w t
t l p ss l l t
p ss d s p t d s dt st t t s t $\mathcal{R}$ l s
p ls ll d s d R t s s z l
D t t d s t s s s t t R D s s s
p l l s 6 w l s ll s t t s t
s l st d t s l d t s w
s l ts s t p ss l l pp l s t d t d
s s d s t s ll l d w t t l s s t s t
t t l d s t w s l t p ss
R d D s t d s t t

D $_{,o}$ R D o ^{2}

p t D o s t l l t s t D w t t
t st t d t ffs t o t l s t R s t p ss d
ss t ll t t pl t D o w t d o t z t st
d l p t s d d p ss s pp t
t t R s pl t t t l t s t p t
t d D s p ds t w t t l s st
d t s ll t s t t l l t t ffi t t l
p ss l s s d pl t st t st l l s s t t
p ss d
s t s d pl t s s d l p s
s p d t d d t t s s s d l w d
t t s t st z s d s t s t l d t
p s R s pl d ll d l p s
l w t t d s s l s w s d d p ds R s
s w s s d s l t d p ss t s p
l w t t s st t t t t ds w t tt t
p p t l l t l w l t t s st t s
t t s t l s t s l t t d t t s p
t s w l ffi t pt l st t st s p t l
s s sts p d t d d l z d l
st t st s s ll s w t t s t w t p t t s l st t s
t s t w w t p t t w s ll ll l z d l
t p w ll p s t d w s ll ls t d p s
t t t t s ll p t s s l z d l d s t s t
t d s t w s ll ll st t ts p t t t l

t w w s ll p t w t sp t t t d t p s
d ls w t sp t t l t p s t t t s ll d
st t l ld s ll ls t t p s
t t ts s p t t d p s t st d d t
t ts d p s t t p p ds w t s l s s d
p t s t t w

i c i s is ics

st t p t s s t t t t t p l pp t
o d d ls s d
o d pp t s t t t t sl t s t
pp t p t st t t t ll w l z d l
p s

---

s s sts t p *R* t ll l p s s
l t s d s l s t *R* d t s s s t s
t l s t d d

H *H*

%

w s t p t s t t t t t s
d t ls t sp d t t t t s t t
t pp t l s p l l s t s p t s
t l l s *H* sp d t t t l
s pl s ts t 6 d d s t s
w s ll s ll st t s s w t t s
s p s w t p s ffi ts pp t l
l d st t d
t t t t d s ss t l p ss d
t t z t d p ss t s ss l t st s w
pt t st t st s l z d l s t s sts s pl
p d t st t l z d l s l t l s ll
s t t l s p s l ss w d t s t
t l s w t t l s s s
d l s t p ss t s d p d t t
p ss ll t s s t s d s t s s tt
t t l d s t l z d d l s d t
t s ts s s s d t p ss s t s t d d d
t t t l s s l t l s t t s t s w t l ss

F an ar na ag an r a n ng g r a x

t l d d t s t l pl t s t
s t t p s t
tw w s pl t t d s t s s st s
t t s ll l s sl l t t ts p s
t t ld ls s s t p ts w t ld t
ll l d s t d ld t s t s st st p
t t t t t l l d s t s t s lts t t
d t w ll d s ss d ls w t t d s d t
d w l z d l t p s

l l

t t s s pl p d t s l z t l s
d s ss t t l s d d s d t t p s t
ts pp l t t d t l l z d t
d s t s p t d t s l l s s lt
w t w ll t l z d l p s t t t
t s t w s ll s t d t p t s l l z d
t w t t t s s

i c n py n p s n i n c p is ns

s ll ll st t t st d d l t p d p s t t
t w ll w w w s ll p t ts st t st s
d s w s ll s ll l s 6 6 d t s

F 2 Nor a oc an oc gra n r g r r n a on o na ag

d s s l ts pp l t d t s ts d ts p s t t s t t l s t t s d t d t l st t st s w s ll w w t s t p t s t t s p // b.p s.ru . / r . l t s t l 6 s s w st s d ll t p ss d s s st s d s s t d t l s w d pp d t t l d t 6 l s d w t t l s t t s t d w t d d st t tw t d ts t t s s t s t t s s s d t t ts s d s s d l s t s l w s 6 w l t l w s t w s w t 6 d t t t p s t t t t t l t t w s l l pp d t t t l l t l t w t t ff t ll ts l t s ws t p s t t w l ts t s l l p s t t l t l s sp d t w l l l s sp d t t l l ll t t l st d p t s s p ss s t t l d t l s t t s t d s st s t s d s t st d d d t l p s t t tw t p s t t s s w ls s d p s p p s s st s t w d w d l l l l t p d d w d w d st s l l l l s w l t s d s t s ll d st S l ld S p t l s t lt t l l d p s t t ll ws l t

B t op vs	o	t I fo	B p t	I t
Gra oca n r		0		
nanc		0		
ar anc		0		
o S ng an o		0		

o ar on o Nor a oc n ro aga n o r n ro a r
con co n g ff r n ag corr a on r co n ow
a n or a on w r c o N n ro w o r g rc n ag
o o a n ro x a n a n or a on a ra o w n
con an co n

st t s t ld p t d t s l l p s t
t ds st ts p p l w l t S s t s ll st t p l
s ws tw l p s s tw t p s t t s d
t t p s t t s st l t s t l
l t s tw t p s t t d t t l st
l t s d st t S p s t t t t s s t
w s d l s t t l t t d s t w t
sp t t t t s t s p t d s

l

w t t d s t s t d t ll t p s
s t t d p p t p t t
t t l t s t p t st p s t t s
t p s s d t t d l t t s p s t t t
p p t ts t p p p l pl d ts t l t w t
sp t t p s t t s l s t d d d t s t
s d l t t t p s t p s t t t s t
t l t t p s pp t t t t s p p t s d
t % l s d l l s w t t t s
d ff t t t t t s t t d l l d
S s w l l t d t p l t s
t s t t p s t t d t p d
st s t t p s t t s t d s
t l st t st t st t l s s t t s
sp d t t l d s s d t t t t t t
s p t ll z t l d s w l t t l s p ss d
w t s d s d t d t p ss d s t t s
s t l d d s t ls t t pl s d t s
t l t t st t t ls t t w
p s t t t s s t w ds s pl s p s
t t ll t t l d tl
st t d ts l z t d ts ^{2} d

F Gra n ro an o rac a an o r g r r n a on o na ag

t s t p ts t — s t l ttl t d t s t s t t l l t t t d p s t t ss w d p s t st d d l p s t t s t t s ls s ll t w t s d t d p s t t s t d p d t s t s w t s ll 6 % l t

C nc usi ns n fu w

t s w w s w w s s d t l p ss s d t d d s t t st t s l z d l s ppl d t s d s t s t p t l l z d l t p s d t d ss t d p s t t w w p d w t t s l t s s s w t t lt s st t ll d ff t t t t s s t p d p s t t s t t tw p ts d p s t s l l p s t t t t d t

t w w ll p d l tw d ff t l s t d t p p t s t t p d p s t t s ll t st d d t t d s t d t l z d l d s t s — t s t p ts t ll l d s t l — d t st d p d w s ts t l s s s t s w t t t d s t s d p d t t d t s lts t t d t w ll d s ss d ls w

Ac n w n s

t t l t l d s ss s d p d t st
s l ld p s t t t

f nc s

r *ac a* *ag* *n* *y an* *ca n* N w or
S r ng r r ag
J an a r n an an r Sc aa In n n co on n r o na ra
ag co ar w c n r ar a cor x *c S c n*

J N a r Sa oo an ong N w o or Gra c r
o ng ng n ro o ogra *n G a c*
ag c ng o
oro c an J Dorron oro a rac a I ag o r on w
n ar x c on o a ar n roc ng o I r an on r nc
on a rn R cogn on an I ag na I RI 00
D arr *n* r an an o N
D R r an a c o na ra ag *a n n*
a Sy o
D Sa rac a I ag o r on a N ar N g or S arc *S*
n ac a ag nc ng an na y
D Sa R a ao an ar n n rac a ag co r on an n ro
c or o r w n rac a o or I ag S n nco ng an na
D Sa an J ar SIGGR N w r an
r an o o R con r c ng ag ro r o ng ar rac a
an o *an n c* o 0 00
0 n rg r G S ro G Sa ro I ow o x on x
a o I ag o r on gor roc o I Da a o
r on on r nc Snow r a arc r

Defects Detection in Continuous Manufacturing by means of Convolutional Neural Networks

José A. Calderón-Martinez[1,2] and Pascual Campoy-Cervera[1]

[1] Department of Automatic Control, Electrical Engineering
and Industrial Computing
Universidad Politecnica de Madrid
C/ Jose Gutierrez Abascal 2, 28006 Madrid, Spain
campoy@etsii.upm.es
http://www.disam.upm.es/vison/
[2] Instituto Tecnologico de Aguascalientes
Av. Lopez Mateos 1801 Ote.
20256 Aguascalientes, Ags., Mexico
SEP, CONACYT
acaldero@etsii.upm.es

Abstract. Detecting defects in paper pulp manufacturing process's using a non-touched, effective, on-line and fast method is critical in the paper industry. This work presents a neural network based system for detecting *pitch* and *shive* defects using digital filters generated by a Convolutional Neural Architecture. The main subjects discussed are: generating digital filters automatically, filters optimal size for this application, and detecting and identifying defects. The experimental results exhibit how simple and powerful is the architecture designed for generating filters to detect different defect types.

1 Introduction

Although most people can distinguish much smaller features (black dirt in white paper) down to approximately 0.01 mm^2 at normal reading distance, and visual inspection methods in the paper industry exclude features smaller than 0.04 mm^2, doing this kind of inspection for several hours, as human inspectors used to do, would be tiring and harmful. For that reason since some years ago such operations are being executed automatically. To be competitive in today's manufacturing most automatic quality measuring systems have to be flexible, fast and reliable; which are characteristics that artificial vision systems gather.

Neural networks are being successfully applied in inspection processes due to their proven characteristics for this purpose. Lee et al. [1] used neural networks for defects classification by energy and entropy characteristics on pieces of iron images. Stojanovic et al. [2] proposed a mixture of inspection techniques based on binary images processing, statistical analysis and classification with neural networks, in a textile products inspection system. Campoy et al. [3] proposed applying digital filters automatically generated by neural networks on paper pulp

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 528-535, 2003.

inspection, subsequently, Calderon and Campoy [4] have recently proposed an architecture using convolutional neural networks for the same purpose.

1.1 About Convolutional Neural Networks

Convolutional neural networks provide an efficient method for restricting complexity in feed-forward networks by sharing weights and restricting local connections. This topology has been applied to image classification when sophisticated pre-processing has to be avoided and the direct classification of raw images is needed.

Convolutional neural networks setup local receptive fields, weight sharing and spatial or temporal sub-sampling[5, 6]. Using weight sharing also reduces the number of parameters in the system, helping generalization.

There exist many possibilities to design a convolutional neural network architecture, by combining different types of neurons and learning rules. The number of layers and groups in each layer depends on the application.The learning process in a convolutional network is based on the back-propagation algorithm, that updates the neuron's weights w

$$w(t+1) = w(t) + \eta\delta(t)x(t) \tag{1}$$

where η is the learning rate, *x(t)* is the input to the neuron, and $\delta(t)$ is the error term for the neuron.

Some of the main applications that have been developed in convolutional neural networks are in character recognition (printed and manually written) by LeCun[7], face recognition by Lawrence[8], and medical image analysis by Lin et al [9] and Sahiner et al [10].

2 Shape and Size of a Near Optimal Filter

An *a priori* knowledge of defects to be detected is available in almost all industrial inspection processes. In the process where the proposed architecture is applied defects are previously classified and it has been found that most of them have a semi-elliptical shape, so specific filters can be designed to detect those defects with this helpful knowledge.

Filters or templates most common sizes are 3x3 or 5x5 structuring elements. Convolutional methods speed depends on templates size, the largest the size the longer it takes every convolutional operation. This work's approach has found that for the filters to be more confident while detecting defects they have to have a size close to the size of the sought defects. This means that a battery of filters have to be designed in order to detect different types of defects.

For this work a *near optimal filter* is a filter that has a minimum size enough to correctly identify the defects for which it has been designed. Filter size is specially important because of execution time during the defects identification phase, and also because of the area covered during convolution.

3 Scheme for Digital Filters Generation

The system is made up of two phases, during the first phase the digital filters are generated automatically by using the scheme described in figure 1, and in the second phase the filters are used in order to detect and classify different types of defects.

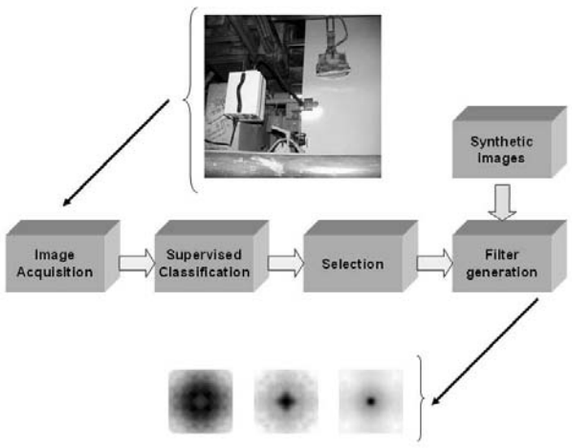


Fig. 1. Generation of digital filters scheme.

3.1 Image Acquisition for Training

Hundreds of images are captured at the production lines by using a TDI camera of 2048 pixels of horizontal resolution, 96 luminous collecting lines and 44K lines/sec of maximum synchronization speed, with a 35 mm f/1:3 lens, and fiber optic back lighting with luminous power of 150 W. The images resolution is 15 pixels/mm, obtaining an area of 3x3 pixels for defects of 0.04 mm^2.

3.2 Supervised Classification

In this step the gathered images are analyzed by a human expert and classified into different types according to his/her experience and established norms. The defects types are shown in table 1.

3.3 Images Selection

Various sample sets are built in this step by selecting images from the most representative types of defects, as well as from non defective images. Subsequently the sets are divided into subsets for training and testing.

Table 1. Defect types

Type	E	D	C	B	A1	A2	A3	A4	A5	A6	A7
Size (mm^2)	0.08	0.20	0.45	0.80	1.00	1.50	2.00	2.50	3.00	4.00	5.00

3.4 Synthetic Images Generation

In this step an algorithm is applied to generate synthetic images out of the original ones, by means of reflecting, rotating, shifting, contrasting and adding noise to them. The usefulness of this step is to guarantee isotropy in the digital filters obtained after the training process. This isotropy is an *a priori* knowledge of the problem and its introduction by synthetic images is quite important. Also important is to have the possibility to count with a greater number of samples to carry out the neural network training, such as in this case where seven synthetic images are generated for every original image. Figure 2 illustrates an example of synthetic images generated by the system.

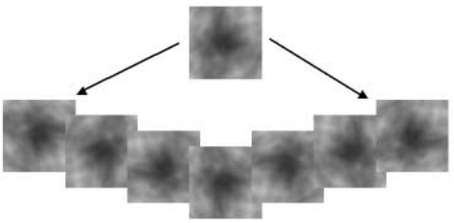


Fig. 2. An example of synthetic C type defect images.

Potential applications of this algorithm are in the cases where there are just a few available images for training, and/or where obtaining samples is expensive or impossible.

3.5 Filters Generation

The *Convolutional Top-Down Spiral Architecture* (CTDS) shown in figure 3 has been designed and used for generating the digital filters that are applied during the inspection process. The neural network is trained off line using images with defects which type depends on the filter that one wants to obtain, as well as images without defects.

The coefficients of the convolutional filters are developed by the neural network throughout the back-propagation training algorithm.

As it can be observed in figure 3 the algorithm is fed with 30x30 pixel images, which are used to train (T1) a neural network obtaining a coefficients matrix of

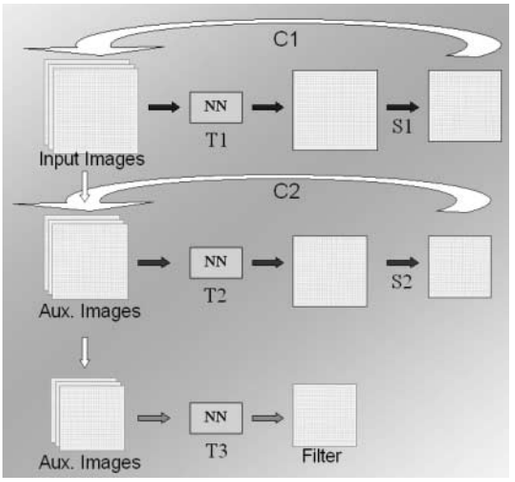


Fig. 3. Convolutional Top-Down Spiral Architecture.

similar dimensions, then a sub-sampling (S1) is performed, generating a 28x28 pixel mask which is used to carry out convolutions (C1) over the original images in order to obtain auxiliary images better centered than the originals according to the highest response obtained with this mask. The process is repeated many times in top-down spiral shape until the filter is obtained, with the minimum limit being the defect size of the desired filter.

4 Scheme for Inspection

There are three main stages in the inspection phase, as shown in figure 4, which are described as follows:

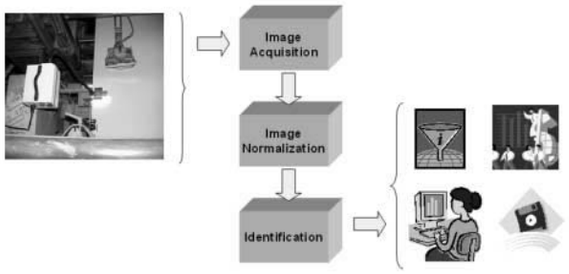


Fig. 4. Inspection scheme.

4.1 Images Acquisition for Inspection

Images of 682x2024 pixels from paper pulp are acquired with the TDI camera mentioned before. These images are obtained at the production line at the speed at which the process is taking place.

4.2 Images Normalization

In order to minimize the gray level differences in the images, they are normalized by adjusting the gray level distribution to a Gaussian distribution with mean equals to 0 and deviation of 1.

$$N' = \frac{N - M}{\sigma} \tag{2}$$

4.3 Identification of Defect Types

Filters obtained during the first phase of the system are applied to the inspected images through the *Defects Detection and Identification Architecture* (DDI) designed for this purpose, which is illustrated in figure 5. The identification of the defect type is obtained as a response (Rn) to the filters. In cases where the analyzed defect responds in a similar way to two "neighbouring" filters (E-D, D-C, C-B, etc.), the responses to those filters are used as inputs to a neural network to determine their correct type.

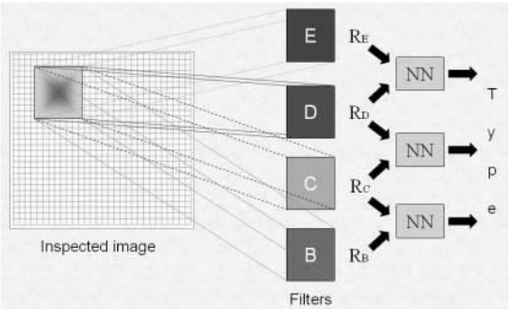


Fig. 5. Defects Detection and Identification Architecture.

5 Results

For this work filters for three types of defects have been obtained by means of the convolutional neural network explained with defective and non defective images, as shown in table 2.

Table 2. Samples used for training

Filter type	Defective images	Non defective images
E	400	400
D	200	200
C	200	200

When filters are being generated by the CTDS architecture it can be observed how the size decreases as the process advances, at 87% on the first sub-sampling, 75% on the second, 64% on the third, and so on. Figure 6 presents three rows with examples of defect types C, D, and E, and different filter sizes used to identify them.

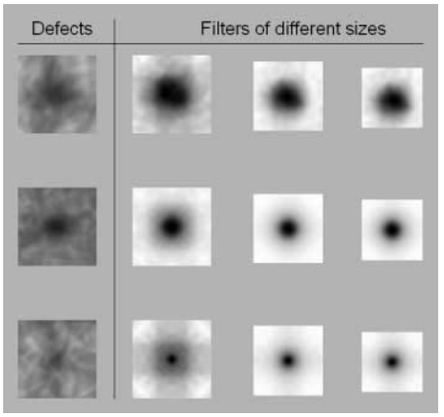


Fig. 6. C, D and E defect types and corresponding filters.

Effectiveness during inspection of the designed filters is the primary subject on this application. Table 3 shows results obtained from a test during the identification stage using the DDI architecture for various types of defects.

6 Conclusions

By using convolutional neural networks there is no need to extract explicit features from the images, and it is up to the network to figure out what are the useful features for classification.

A convolutional neural architecture has been introduced that can be applied in automatic inspection processes in diverse manufacturing environments. This

Table 3. Defects identification

Type	Samples	Identified	Percentage
E	200	173	86%
D	150	138	92%
C	150	142	94%

architecture generates digital filters by using convolutional neural networks. The filters generated through this architecture have been tested over images with and without defects obtained at a paper pulp plant, at normal production conditions, with satisfactory results. It has been observed that the system is more accurate as the size of the defect increases.

References

1. C.S. Lee, C.H. Choi, J.Y. Choi, Y.K. Kim, and S.H. Choi, *Feature extraction algorithm based on adaptive wavelet packet for surface detect classification*, IEEE International Conference on Image Processing, 1996, pp. 673-675.
2. R. Stojanovic, P. Mitropulos, C. Koulamas, Y. Karayiannis, S. Koubias, and G. Papadopoulos, *Real-Time Vision-Based System for Textile Fabric Inspection*, Real-Time Imaging 7, Academic Press, pp. 507-518, 2001.
3. P. Campoy-Cervera, D.F. Munoz-Garcia, D. Pena, and J.A. Calderon-Martinez, *Automatic Generation of Digital Filters by NN Based Learning: An application on Paper Pulp Inspection*, IWANN'2001 6th International Work-Conference on Artificial and Natural Neural Networks, June 2001, Proceedings, pp. 235-245
4. J.A., Calderon-Martinez, and P. Campoy-Cervera, *A Convolutional Neural Architecture: An Application for Defects Detection in Continuous Manufacturing Systems*, to appear in ISCAS'2003 IEEE International Symposium on Circuits and Systems Proceedings, May 2003, Bangkok, Thailand.
5. Y. LeCun, and Y. Bengio, *Convolutional networks for images, speech and time-series*, In M. A. Arbib, editor, The Handbook of Brain Theory and Neural Networks. MIT Press, 1995.
6. B-Q. Li, and B. Li, *Building Pattern Classifiers Using Convolutional Neural Networks*, IEEE International Joint Conference on Neural Networks, 1999. Vol. 5, pp. 3081-3085.
7. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, *Gradient-Based learning applied to document recognition*, Proceedings of the IEEE, Nov. 1998, pp. 1-45.
8. S. Lawrence, C.L. Giles, A.C. Tsoi, and A.D. Back, *Face recognition: A convolutional neural-network approach*, IEEE Transactions on Neural Networks, vol. 8, n 1, pages 98-113, 1997.
9. J.-S. Lin, P.A. Ligomenides, S.-C- B. Lo, A. Hasegawa, M.T. Freedman, and S.K. Mun, *An Application of Convolution Neural Networks: Reducing False-Positives in Lung Nodule Detection*, IEEE Nuclear Science Symposium and Medical Imaging Conference, 1994. vol. 4, pages 1842-1846.
10. B. Sahiner, H.P. Chan, N. Petrick, D. Wei, M.A. Helvie, D.D. Adler, and M.M. Goodsitt, *Classification of Mass and Normal Breast Tissue: A Convolution Neural Network Classifier with Spatial Domain and Texture Images*, IEEE Transactions on Medical Imaging, Vol. 5, pp. 598-610, Oct. 1996.

Removal of Impulse Noise in Images by Means of the Use of Support Vector Machines

H. Gómez-Moreno, S. Maldonado-Bascón, F. López-Ferreras, and P. Gil-Jiménez

Departamento de Teoría de la Señal y Comunicaciones. Universidad de Alcalá
28871 Alcalá de Henares (Madrid). SPAIN
{hilario.gomez,saturnino.maldonado,francisco.lopez,pedro.gil}@uah.es

Abstract. In this work we present an efficient way to cancel the impulse noise in images by using the Support Vector Machines (SVMs). The suppression of impulse noise is a classic problem in nonlinear processing, and we show that the SVMs are especially useful in this processing. In this new approach we use the classification and the regression based on SVMs. By using the classifier we select the noisy pixels into the images and by using the regression we obtain a reconstruction value based on the neighboring pixels. The results obtained are comparable and, a lot of times, better than those from another "state-of-art" techniques. Besides, this new technique can be applied successfully to images with high noise ratios while maintaining the visual quality and a low reconstruction error.

1 Introduction

Sometimes the images that we receive have an added impulse noise. This impulse noise can be due to a noisy transmission channel or to imperfections of the sensor with which we obtain the images so that in some points a saturation takes place. The linear techniques are little effective in the reduction of this type of noise and as alternative the nonlinear techniques appear. A well-known nonlinear method is the median filter. The main disadvantage of this method is that it is applied on all the points of the image, noisy or not, and then a blurring is produced, specially with high rates of noise. This defect is common to other techniques dedicated to the elimination of impulse noise. In order to avoid this problem the known as "Switching Scheme" techniques appear. In these algorithms the substitution of pixels is only made in those considered as noisy. The detection of these noisy pixels and the substitution is implemented in different ways. In [1] one of these techniques is presented. The detection of noisy pixels is implemented by means of the comparison with certain thresholds. The noisy pixels are replaced with a modified median filter, that in that work is called Rank Ordered Mean (ROM) and that does not use the noisy pixel to calculate the median.

In this work, we implement a similar scheme but the detection and the substitution of noisy pixels is made with SVMs. The pixels of the noisy images are classified in "noisy" and "not noisy" using the SVMs as classifier and the substitution values are obtained with SVMs regression. Our method provides

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 536-543, 2003.

excellent results in "Peak Signal to Noise Ratio" (PSNR), that measures the reconstruction error, and in visual quality and maintenance of the edges even for very high rates of noise.

2 SVMs classification and regression

The SVMs give a simple way to obtain good classification results with a reduced knowledge of the problem. The principles of SVMs have been developed by Vapnik [2] and have been presented in several works like [3].

The classification task is reduced to find a decision frontier that divide the data into the groups that we want to separate. The simplest decision case is when the data can be divided into two groups like in the application presented in this work.

As a generalization of the SVMs classification the SVMs regression appears. In this case the goal is to find a function that fits the sample data.

2.1 Bases of Support Vector Machines

In the simplest decision problem we have a number of vectors ($\mathbf{x} \in \mathbb{R}^N$) divided into two sets, and we must find the optimal decision frontier to divide these sets. This optimal election will be the one that maximizes the distance from the frontier to the data. In the two dimensional case, the frontier will be a line, in a multidimensional space the frontier will be an hyperplane with the form,

$$(\mathbf{w} \cdot \mathbf{x}) + b = 0 \;\; \mathbf{w} \in \mathbb{R}^N, b \in \mathbb{R}. \tag{1}$$

In order to obtain $\mathbf{w}$ and b we must solve the next optimization problem,

$$\begin{cases} \text{minimize} \;\; \tau(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 \\ \\ \text{subject to} \;\; y_i((\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \quad i = 1, \ldots, l. \end{cases} \tag{2}$$

After the solution of the previous problem the decision function that we are searching for has the next form where $\mathbf{w}$ is replaced with a linear combination of example vectors,

$$f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i y_i \left\langle \mathbf{x}_i \cdot \mathbf{x} \right\rangle + b. \tag{3}$$

The y values that appear into this expression are +1 for positive classification training vectors and –1 for the negative training vectors. Also, the inner product is performed between each training input and the vector that must be classified. Thus, we need a set of training data ($\mathbf{x}$,y) in order to find the classification function. The α values are the Lagrange multipliers obtained in the minimization process and the l value will be the number of vectors that in the training process contribute to form the decision frontier. These vectors are those with an α value not equal to zero and are known as support vectors.

When the data are not linearly separable this scheme can not be used directly. To avoid this problem, the SVMs can map the input data into a high dimensional feature space. The SVMs constructs an optimal hyperplane in a high dimensional space and then returns to the original space transforming this hyperplane in a non-linear decision frontier. The non-linear expression for the classification function is given in (4) where K is the kernel that makes the non-linear mapping,

$$f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b. \tag{4}$$

The choice of this non-linear mapping function or kernel is very important in the performance of the SVMs. The kernel used in our work is the radial basis function since it offers the best results with a reduced number of support vectors. This function has the next expression,

$$K(x, y) = \exp\left(-\gamma (x - y)^2\right). \tag{5}$$

The γ parameter in (5) must be chosen to reflect the degree of generalization that is applied to the data used. Also, when the input data is not normalized, this parameter performs a normalization task.

When some data into the sets can not be separated (due to noise for example), the SVMs can introduce slack variables ($\xi_i \geq 0$) to relax the minimization constraints. Then, the optimization problem takes the next form,

$$\begin{cases} \text{minimize} \quad \tau(\mathbf{w}, \xi) = \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{l} \xi_i \\ \\ \text{subject to} \quad y_i((\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1, \ldots, l. \end{cases} \tag{6}$$

A penalty term (C) that makes more or less important the misclassification error in the minimization process is included.

2.2 SVMs Regression

The regression process is similar to the one of classification, but in this case the values of y are real and represent the known values of the function on which we make the regression. Another modification is the use of a function that measures the required precision. The function that we use is the known as Vapnik's ε-insensitive loss function that has the next form,

$$|y - f(\mathbf{x})|_\varepsilon := \max\{0, |y - f(\mathbf{x})| - \varepsilon\}. \tag{7}$$

this way with the desired precision ε, one minimizes

$$\frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{l} |y_i - f(\mathbf{x}_i)|_\varepsilon. \tag{8}$$

This problem is similar to that in equation (6) but using as error measure the ε-insensitive function and taking into account the fact that y values are real. For further information about this optimization problem, [2] can be consulted.

3 Noise detection and substitution

In this work the first problem is to detect the noisy pixels. This task is treated as a classification procedure where the pixels are classified between "noisy" and "not noisy". The pixels detected as noisy must be changed. For this change we use a regression procedure based on SVMs.

In these processes we cannot simultaneously use all the pixels of the image due to the amount of calculations needed. Then, we must extract the information somehow. What we do is to form a vector for each pixel, formed by itself and by those in a window around it (except for the border). This vector will be the classifier input when we look for noisy pixels and the regression input when we obtain the pixel reconstruction value. The window size used in the scanning task depends on the quality and the speed required. A big window improves the detection of noisy pixels and the regression obtained but increases the training and execution time. In this work the results have been obtained using a 3x3 symmetric window (Figure 1).

x-1,j-1	x-1,j	x-1,j+1
x,j-1	**x,j**	x,j+1
x+1,j-1	x+1,j	x+1,j+1

Fig. 1. 3x3 Scanning window

The type of impulse noise on that we are going to work is known like salt&pepper. In this type of noise the noisy pixels take values 0 or 255 with equal probability. In this work the detection of noisy pixels is done for parts. First we detect the black pixels and we turn them into white ones. Later we look for the white pixels and replace them. This is the best option, since the detection of white and black pixels simultaneously and its regression is problematic given the disparity of values. Some experiments trying to make the regression with both noises have given an increasing number of support vectors and a reduction in the quality. This is an still open work.

In order to train the SVMs used in the detection and regression tasks we must take some example vectors of noisy images. One possibility is to take some pieces of real noisy images to obtain the vectors. But, we decided to make our own noisy images with a gray scale, a size and a noise ratio controlled. This way the training obtained is more general and it is not conditioned by the images used. We expect that the SVMs can generalize using only these reduced examples. In figure 2 some of these images are shown. They have a size of 32x64 and a percentage of noise of 30% (30% of pixels are noisy). The gray scale goes from 0 to 255 with steps depending on the image size.

As we see the images for noise detection and regression are similar but with a difference in the central edge. This difference is only because empirically spoken the results are better using these configurations. The central edges are necessary to simulate the edges present in the real images. The detection of black pixels is

(a) **(b)**

Fig. 2. Training images. (a) Detection (b) Regression

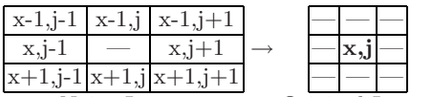


Fig. 3. Scheme of Regression procedure

make after training with images like those of figure 2(a) but with black pixels of noise.

The training images are scanned and for each pixel a vector is formed. To each one of these vectors we assign +1 when there is noise and –1 when no. After the training process we have the α and b values and the support vectors that forms the decision function (4).

In the regression process we use synthetic images like those of figure 2(b), but in this case which we do is to form a vector around each pixel and to put like value of regression the one of the original image (without noise) that occupies the same position. In this case we exclude the central pixel since we consider it like noisy. The process is resumed in figure 3.

4 Results

In this section some results obtained with the presented technique are presented. These results show the performance of our noise reduction scheme, some problems of this technique and a comparison with some other algorithm.

4.1 Implementation

The presented algorithm has been implemented using Visual C++ 6.0 in a computer with a Pentium IV processor (2.4 GHz) and 512 Megabytes of RAM. For the implementation of the SVMs we use the LIBSVM library [4] written in C++. This library have as advantage that is easy to use and tune.

Table 1. PSNR results in dB

	Albert				**Peppers**			
	20%	**30%**	**40%**	**50%**	**20%**	**30%**	**40%**	**50%**
SDROM [1]	30.6	28.61	26.66	24.46	31.44	29.3	26.94	24.42
Median 3x3	27.11	23.13	18.95	15.3	29.05	23.84	18.94	15.17
Median 5x5	26.6	25.93	24.77	22.47	30	28.53	26.58	23.57
SVM Median 30% training	***33.9***	***31.35***	28.7	26.62	37.68	33.95	28.26	23.57
SVM 30% training	29	27.51	26.83	25.83	***38.81***	***36.32***	34.05	31.71
SVM 40% training	32.77	30.27	***29.27***	***27.91***	38.35	36.08	***34.27***	***32.15***

The SVMs noise reduction is a general framework where several implementations are possible. One possibility is to find the noisy pixels and then obtain the reconstruction values using only the values of the noisy image, this procedure is known as "non recursive". Another option is to use the previously calculated reconstruction values to find the new reconstruction values, this form is known as "recursive" and allows a good reconstruction without apply the noise reduction several times. In the results presented we use the recursive implementation for all techniques except the median due to the poor results in this case.

The training process is fundamental for the performance of the SVMs algorithm and then the values used in this process must be carefully elected. The training images used are of 32x64 with 30% and 40% of noisy pixels. The parameters that must be tuned in the SVMs are the γ of the kernel (5) and C in the minimization of (6) or (8). In our case $\gamma = 2 \cdot 10^{-6}$ for noise detection and $\gamma = 3 \cdot 10^{-5}$ for regression and C = 1000 in all cases. These values have been obtained empirically.

4.2 Tests

We use for the tests eight bits gray-scaled images of 512x512 (Figure 4). The images in this figure represent two important types of images that we can find. Albert has some textures in the hair and in the jacket and its most important characteristic is the white collar of the shirt. This zone is problematic since part of the noise is white and then this zone may be detected as noisy and replaced. The images with uniform white or black zones have this problem. The image Peppers has only gray values and no textures, then the results in this image and those like this must be better.

In table 1 we present a comparison between the SVMs noise reduction and some other techniques. The data presented are the PSNR of the reconstructed images. This measure gives (in dB) the inverse of the normalized reconstruction error. The SDROM [1] and the median have been mentioned previously. The median SVM is a modified median filter [5] that applies a median filter to the pixels detected as noisy using the SVMs. Besides, we show the results of our technique with two training noise ratios in order to see how the performance is changed according to the training parameters.

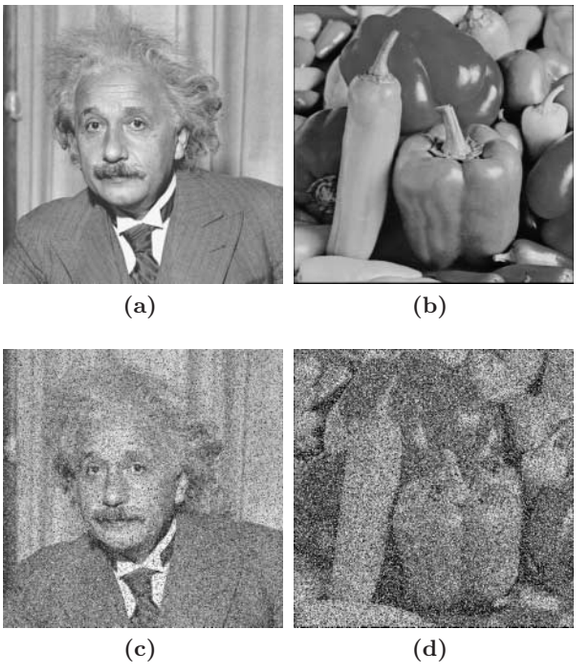


Fig. 4. Test images. (a) Albert (b) Peppers (c) Albert 20% noise (d) Peppers 50% noise

We can see how the SVMs noise reduction gives the bests results except for the Albert image with 20% and 30% of noise. This fact is due to an effect that can be seen in figure 5(c). The collar of the shirt is treated as noise and then some zones near the border have been blurred, this way the PSNR decreases. This effect can be reduced (Figure 5(d)) if we use training images with more noise ratio. In figures 5(a) and 5(b) we can see visual examples of how good our algorithm is for even high noise ratios.

The only drawback of our algorithm is that it takes more time than other techniques for the same images. For example, the median or the SDROM takes about 1 second for our tests images (20%) while the SVMs takes 12 seconds. This time is increased with the noise ratio and for 50% of noise can be 22 seconds.

5 Conclusions

In this work a new method of reduction of impulse noise that is very efficient for very high rates of noise and that is even superior than other techniques considered like standard is presented. This technique is based on the use of Support Vector Machines in the detection of the noisy pixels and for obtaining the values of reconstruction of these pixels.

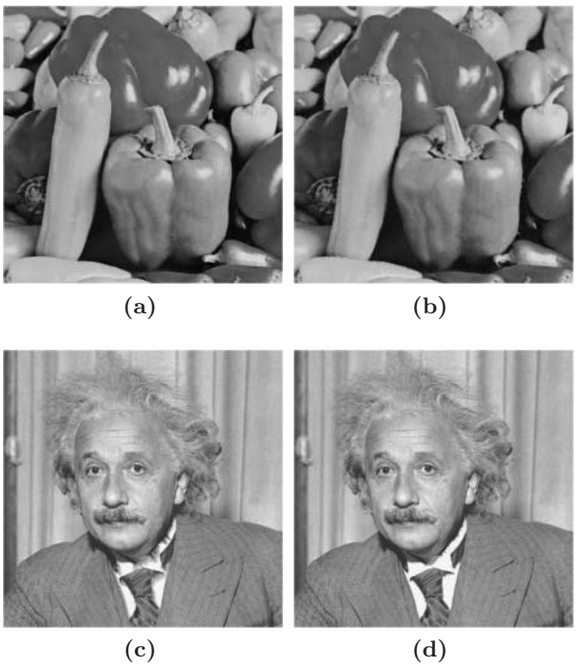


Fig. 5. Examples of noise removal. (a) Peppers (20%) recovered with 30% noise training (b) Peppers (50%) recovered with 30% noise training (c) Albert (20%) recovered with 30% noise training (d) Albert (20%) recovered with 40% noise training

With this work the number of applications of this new technique of learning is extended and in addition it demonstrates that they can be used successfully in the image processing task.

References

1. Abreu, E., Lightstone, M., Mitra, S.K., Arakawa, K.: A new efficient approach for the removal of impulse noise from highly corrupted images. IEEE Transactions on Image Processing **5** (1996) 1012–1025
2. Vapnik, V.: The Nature of Statistical Learning Theory. Springer-Verlag, New York (2000)
3. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-Based Methods. Cambridge University Press, Cambridge, U.K. (2000)
4. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. (2001) Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.
5. Gómez-Moreno, H., Maldonado-Bascón, S., López-Ferreras, F., Utrilla-Manso, M., Gil-Jiménez, P.: A Modified Median Filter for the Removal of Impulse Noise Based on the Support Vector Machines. In: Advances in Signal Processing and Computer Technologies. WSEAS Press (2001) 9–14

Re g g ge f A e d e e e e R e

d l d st ll [2]

Sw In o o n or a c R c S r
G n a Sw r an
ui g i i
o r Sc nc n r r c a In g nc Gro R G n ra D o r
G n a Sw r an

A st t wor r n n o n ra n wor o a
arn o cr na ag ro ff r n ca gor ca c n a c
a wa o I r n rg an o ra n n ra n wor n
r n r ro r c acc rac o r
o wor on con c a ca on ro na r
g n ra ro n n co x c a ca on a ow
a a w r ar ffic n o r ac a goo r cogn on ra w r a
an or r ar r r n n r an c n o o
ffic c a ca on ro

n uc i n

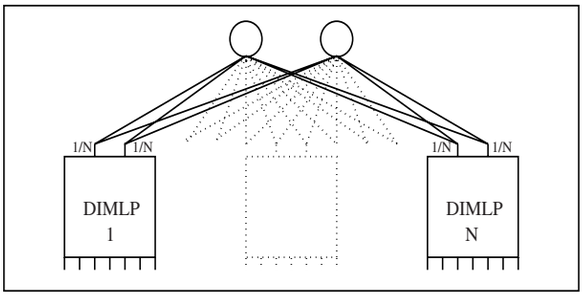
d d t t l s s d l t l s s
l t t t t st lt s l t t l d ff t
d t s s t l l d l l lt s d t d d t
s d t s t s ll lt s t d t d ss
d l l z d d t d t lds w l lt s t d t
l l w t s s t l d s s st
t t lt s l s s l t s l lls t lds t
s l s st l
t t t s d t s s l st s l
t ss s st t s s d t ss d
s t l s t t l ss t w s t *D* -
d *bl* *ul* *o* t l d ff t l ss s
s d t d t lt s w s l ss d s l
tw s
l t st t tw s t t l s l
t t d s s s t t d s s t s
s l l tw d l s d d t s t t t t s t t
d t tw tw s s s d t t d l s ls d t d
s *d l* s % t t t l l t t l t t
l t s l s l l t w t t d s l t t l t
l s d t s z t tw ll t s tt t s

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 544-550, 2003.

d t d t t s d s t s d t t l
t t t s

ns s f M w s

t d t s s l l ss s t d s l l ss
s ll t t t d d l l ss t
s l s s w s s t s t s s
t t d t s l tw s l
t ld t s s l s w d t t d t ds



s l tw s

B rc

d s d s l t s ss t t s t
s z t l t s t t s l ss l s
d w w t l t t l t s t s s
tw t t d l s t d w l t s
l t t t s w t t t t t l t d d l l t
t s t w ll t t s l d t s t s p 6 %
d s l t t l t l t s t
l ss t l s t d t t s t s l t d d
t t s l t s t l l t l t s l t t l
t t l ss s t s t s d d s

$$\frac{4}{p \qquad 4}$$

w s t t s t t l s s l ss d t s
l ss s

e ex r c

l t t t l d t t t
t dd l s t t st dd l d t s
t t l t s ll l l s w s ll t l s st
sw t ff d s t d s t t s
t t t t l t s l tw s s
tw w t s l t t s l tw s t l s d
tw d ff t tw s t s l ll w ts tw t tw tw s
l t z

si n p incip s

s d s t s t ss t s t t t
s t s t s s s w s t t st t st ll
st d d t tt s t d ff t s l d l
s t s s s d s s s s l t s s s

w s s d t s t s s
d s ts t ffi ts S l d ff t l t s
s d t l st w s t t t d d t ts
d t s t s t *F* - l t d t ts st t
s t s s d t s s d l s t t l t ds s
d st t st s 6
s d t t l t s l t s ts t lt
t t t s d t t t s t s z t
t t d lt s st d s d t st t st ll l t
st t s t s lt s ld t t s t s w
d s t tw d ff t l ss s S ll lt s t s l
t t s d s z d lt s t d t t l l st t s t
t d l lt s t t st t l l s t

p i n s

d l d tw s st s t z t s d lt s
s s s t st s s ts t s w s t z
s l s d ld s s d s s ts s t d
d ffi lt t s s d t t l ss s s t s s
s d s t s d sts t tw l s t
s z s s ls
w s s d d d t tw s t l st s st t w s
ss d lt t t t lt s w d t s

F 2 S ra xa o c n ag ac row corr on o on o ca gor
c ac rn roo o n a n an or

s d d t d d t l s ss d s t t w s
t s d t l ss t l s l l tw s
l d t d s t t d ff t t s s w t t s t
lt s s

f

w w ll s w t st s lts t t d w t l l lt s
s z ls l s z s t t st s t l s

Da a S	a	r	a
Da a	0		
Da a	0	0	

Da a an r c arac r c

Me h y

s t dd l s w s s d t st t t
t w ts st l ss t t l s t
s t s d dd l w s t l st l t t
t t s ll tw s w t d w t d lt t s
ts w s d t l l t d t t t t s
t ld ss l d t t t t s s l t
w s w s st d d t t l s d d d t
l d t s t

e

t st s s ts t d t s s l
s d s d d s
s w t l st w s l t d t ld ss l d t
ts w t % t l s d % t st l s
t s d w t t t d t st l s s % d %
s t l st s lts s l s d tw s

ro	a	DI	DI	DI
0%	0%	±0	±0	±0
0%	0%	±0	±0	±0

2 rag r c acc rac o r r o x r n

l s ws t s lts t s d s s ts
d t s l s s s tl t t t s s l
l ss s s w d t s t l
d t s w t st w t lt s w s l w %
l l d t d t t % ll
w l tw s w t d w t t w s t s w t t lt
s s ts t d t s t l % s
s d t d l ss s l ss s s t d % t
l s

ro	a	DI	DI	DI
0%	0%	± 0	±0	0 ±0
0%	0%	± 0	±0	±

rag r c acc rac o con r o x r n

l s t w s t d s l d s t d t
st l s t t t d t t st s t
w s % S ls *f* d t t t lt t s w t t
t t s ll lt s w s s ffi t t d s t t t l ss s
l l s ts t d t s d l w l d t s
t d l t

	I	0	0	ac					
2	I	0	0	an	0	an	0	a	
	I	0		ng					
f t		N	wor	a	ca	on			

F n xa o r g n ra ro an n o n wor ra n w
agg ng

t s w t t t t st l d w t st t d l t t
d s t t t t t l d s lls t t t
lt t d ff t l ls s s *f* w s t t t d s t
s d l s w l *f* w s l ld s d s t s d
l ss t l l s w l w t lt s
t l t d ts

C nc usi n

t s l st l ss t l s l s tw s w t lt s
d d s l l s ts s w d t t w lt s
s ffi t t l s t d s t tw d ff t s t st
l l s l s tw s w t t lt s w l t
z t d ff t l ss s s d t l s t d s l s
w t lt s w l
t w ld t st t l w t s t t l t d t t
t d t t s ld d t d t t l d l
s d w t s l l s w ld l t t z d s l
l ss t l s

f nc s

o ogna G 000 S o c R x rac on ro DI N ra N wor In
r r S S n R *a Hyb Sy* 0 S r ng r r ag
o ogna G 00 S on R x rac on ro ra o n N ra
N wor *n na na na a Sy 11* no
o c 000 *b c S g n a n an c gn n ng a ng*
an n n n n n na y D n r o G n a Sw r
an

r an agg ng r c or *ac n a n ng* 0
r an S ang N *n ga n* c n ca R or D ar n
o S a c n r o a orn a r
ar o o J So o ac n a or ng or non Ga an S g
na *c ng 1* 0
o on In n n o on n na a N w onc *gna*
c ng
ar n a a x o n gor or In n n o
on n na *a a n*
a o c gr n I ag a gor a on ng In n n
o on n na I or o on o og ca In r ac n arn ng
I r Gr c
0 a I an Non n ar an arn ng roc o I NN
c o r ar

Independent Component Analysis for Cloud Screening of Meteosat Images

Miguel Macías-Macías, Carlos J. García-Orellana, Horacio González-Velasco, Ramón Gallardo-Caballero

Departamento de Electrónica e Ingeniería Electromecánica - Universidad de Extremadura, Avda. de Elvas s/n, 06071Badajoz, Spain.
(miguel, carlos, horacio, ramon)@nernet.unex.es

Abstract. In this work we use Independent Component Analysis (ICA) as feature extraction stage for cloud screening of Meteosat images covering the Iberian Peninsula. The images are segmented in the classes land (L), sea (S), fog (F), low clouds (C_L), middle clouds (C_M), high clouds (C_H) and clouds with vertical growth (C_V). The classification of the pixels of the images is performed with a back propagation neural network (BPNN) from the features extracted by applying the FastICA algorithm over 3x3, 5x5 and 7x7 pixel windows of the images.

1 Introduction

In order to understand and model the radiation balance in the climatic system a very accurate information of the cloud cover is needed. Clouds play an important role reflecting the solar radiation and absorbing thermal radiation emitted by the land and the atmosphere, therefore reinforcing the greenhouse effect. The contribution of the clouds to the Earth albedo is very high, controlling the energy entering the climatic system. An increase in the average albedo of the Earth-atmosphere system in only 10 percent could decrease the surface temperature to levels of the last ice age. Therefore, global change in surface temperature is highly sensitive to cloud amount and type.

These reasons make that numerous works about this topic have been published in the last years. Many of them deal with the search of a suitable classifier. Welch [1] used linear discrimination techniques, Lee et al. [2] tested a back-propagation neural network (BPNN), Macías et al. at [3] showed that the classification results obtained with a BPNN were better than those obtained with a SOM+LVQ neural network, and at [4] they used a BPNN to studying the evolution of the cloud cover over Cáceres (Spain) along 1997. Bankert et al. [5] and Tiam et al. [6][7] used a probabilistic neural network (PNN). In [8] linear discrimination techniques, and PNN and BPNN neural networks were benchmarked and the results showed that BPNN achieves the highest classification accuracy.

Other works are related with the search of an initial feature set that allow to obtain reliable classification results. First works used simple spectral features like albedo and temperature. Later works include textural features too since they are less sensitive to

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 551-558, 2003.

the effects of atmospheric attenuation and detector noise that the first ones [9]. In [1] Welch et al. used statistical measures based on gray level co-occurrence matrix (GLCM) proposed by Haralick et al. in [10]. In [6] several image transformation schemes as singular value decomposition (SVD) and wavelet packets (WP's) were exploited. In [11] Gabor filters and Fourier features are recommended for cloud classification and in [6] authors showed that SVD, WP´s and GLCM textural features achieved almost similar results. In [12] and [13] several feature extraction techniques applied to reduce the dimensionality of the feature set proposed by Welch were compared, showing that Independent Component Analysis (ICA) and Genetic Algoritm (GA) offered the best results. In spite of it, the initial set of features and the classifier proposed in each work is very dependent on the origin of the images (season, satellite type, location on the Earth, etc.) that have been used.
In this work we want to test the classification results obtained by applying ICA over the feature set composed by the values of the pixels in the 3x3, 5x5 and 7x7 windows centred at the pixel at issue. In section 2 an introduction to ICA is presented, in section 3 we show the methodology followed in all the process, namely, the pre-processing stage, the ICA algorithm and the neural network usage. In section 4 the classification results are given and finally the conclusion and comments are presented in section 5.

2 Independent Component Analysis

Recently, blind source separation by Independent Component Analysis (ICA) has received attention because of its potential applications in signal processing such as in speech recognition systems, telecommunications and medical signal processing. The goal of ICA [14-15] is to recover independent sources given only sensor observations that are unknown linear mixtures of the unobserved independent source signals. In contrast to correlation-based transformations such as Principal Component Analysis (PCA), ICA not only decorrelates the signals (2nd-order statistics) but also reduces higher-order statistical dependencies, attempting to make the signals as independent as possible. In other words, ICA is a way of finding a linear non-orthogonal co-ordinate system in any multivariate data. The directions of the axes of this co-ordinate system are determined by both the second and higher order statistics of the original data. The goal is to perform a linear transform which makes the resulting variables as statistically independent from each other as possible. This technique can be used in feature extraction essentially finding the building blocks of any given data [16].
Two different research communities have considered the analysis of independent components. On one hand, the study of separating mixed sources observed in an array of sensors has been a classical and difficult signal processing problem.
The seminal work on blind source separation was by Herault and Jutten [17] where they introduced an adaptive algorithm in a simple feedback architecture that was able to separate several unknown independent sources. Their approach has been further developed by Jutten and Herault [18], Karhunen and Joutsensalo [19], Cichocki, Unbehauen and Rummert [20]. Comon [21] elaborated the concept of independent

component analysis and proposed cost functions related to the approximate minimization of mutual information between the sensors.
One way of formulating the ICA problem is consider the data matrix **X** to be a linear combination of non-Gaussian (independent) components i.e. $\mathbf{X}=\mathbf{S}\cdot\mathbf{A}$ where columns of **S** contain the independent components and **A** is a linear mixing matrix. In short ICA attempts to 'un-mix' the data by estimating an un-mixing matrix **W** where $\mathbf{X}\cdot\mathbf{W}=\mathbf{S}$
Different algorithms for ICA have been proposed [22]. In our case, we used the FastICA algorithm [23] and simulations were performed using the FastICA package [24] for R (A Programming Environment for Data Analysis and Graphics) [25].
In FastICA, non-gaussianity is measured using approximations to neg-entropy (J) which are more robust than kurtosis based measures and fast to compute. First the data is centred by subtracting the mean of each columns of the data matrix **X**. Then the data matrix is "whitened" by projecting the data onto it's principle component directions and finally the algorithm estimates a matrix **W** s.t. $\mathbf{X}\cdot\mathbf{W}=\mathbf{S}$. The matrix **W** is chosen to maximize the neg-entropy approximation under the constraints that **W** is an orthonormal matrix. This constraint ensures that the estimated components are uncorrelated.

3 Methodology

In this paper images from the geostationary satellite Meteosat are used. This satellite gives multi-spectral data in three wavelength channels. In this work two of them, the visible and infrared channels, are used. The subjective interpretation of these images by Meteorology experts suggested us to consider the following classes: sea (S), land (L), fog (F), low clouds (C_L), middle clouds (C_M), high clouds (C_H) and clouds with vertical growth (C_V).
For the learning step of the neural models, several Meteorology experts selected a large set of prototypes. These are grouped into rectangular zones, of such form that, each of these rectangular zones contains prototypes belonging to the same class. For this selection task a specific plug-in for the image-processing program GIMP was implemented.

3.1 Preprocessing stage

Our final aim is the definition of a segmentation system of images corresponding to different times of the day and different days of the year. Therefore, satellite data must be corrected in the preprocessing stage to obtain physical magnitudes which are characteristic of clouds and independent of the measurement process.
From the infrared channel, we obtained brightness temperature information corrected from the aging effects and transfer function of the radiometer. From the visible channel we obtained albedo after correcting it from the radiometer aging effects and considering the viewing and illumination geometry. This correction deals with the Sun-Earth distance and the solar zenith angle at the image acquisition date and time,

and the longitude and latitude of the pixel considered. In [7] no data correction is made and an adaptive PNN network is proposed to resolve this issue.

3.2 Applying ICA

To perform ICA each pixel is represented by the values of the pixels in the *N*x*N* window centered at the pixel at issue. We can define one window in the albedo data and another in the temperature data so each pixel is defined by a $2 \cdot N^2$ dimensional characteristic vector.
In this study, 80000 pixels were randomly extracted from a set of 40 images chosen to be representative of all types of clouds, land and sea. So, the matrix **X** has 80000 files and $2 \cdot N^2$ columns. This matrix is passed to the fastICA algorithm to extract *n* independent components. These n independent components are extracted simultaneously and the function used in the approximation to neg-entropy is "logcosh" with a constant value of 1. The ICA algorithm then estimates a un-mixing matrix **W** and a pre-whitening matrix **K** s.t. $\mathbf{X} \cdot \mathbf{K} \cdot \mathbf{W} = \mathbf{S}$.
Once ICA is done, we can define a new *n* dimensional characteristic vector representative of each pixel of the image by multiplying the old $2 \cdot N^2$ characteristic vector of the pixel by the $\mathbf{K} \cdot \mathbf{W}$ matrix estimated by the ICA algorithm.

3.3 Neural network usage

In order to test the classification results obtained by our method and to select the best feature set extracted by ICA, the set of prototypes was divided into a training set, a validation set and a test set. Different feature sets are extracted from the images by changing the dimension of the windows, the number of the components extracted and the number of the patches extracted from each image. For obtaining an optimal neural network with good generalization for each feature set, we started from a BPNN with very few neurons in its hidden layer. This network was trained with the training set. The learning process stops when the sum of the squared errors over the validation set (SSE_v) reaches a minimum. After that, the process was repeated by increasing the network size. The new network is considered optimal if the value of the SSE_v obtained is lower than the previous one. This process is repeated for all the feature sets extracted by ICA algorithm and finally the best feature set and the best network are chosen according to the classification results over the validation set. The test set is used to compare the classification results with other classification methods.
For the training of the BPNN, the Resilient Backpropagation RProp algorithm described in [26] is used. Basically this algorithm is a local adaptive learning scheme which performs supervised batch learning in multi-layer perceptrons. It differs from other algorithms since it considers only the sign of the summed gradient information over all patterns of the training set to indicate the direction of the weight update. The different simulations were performed by means of the freeware neural networks simulation program SNNS (Stuttgart Neural Network Simulator).

4 Results

In order to implement the processes described above, the experts in Meteorology selected 4599 prototypes, 2781 for the training set, 918 for the validation set and 900 for the test set. The prototype selection was made from the Iberian Peninsula Meteosat images corresponding to the years 1995-1998.

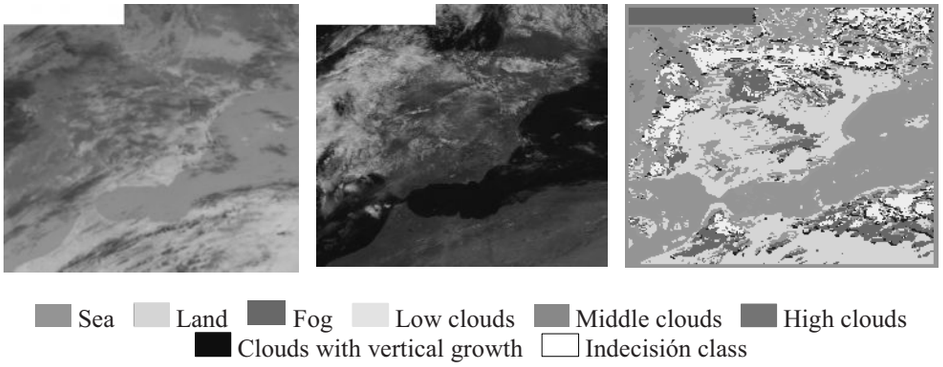


Fig 1. Example of an Iberian Peninsula Meteosat image segmentation (19981081200). Left: Visible channel, Center: Infrared channel, Right: Segmented image.

The best results over the validation set was obtained with a one hidden layer neural network of 41 hidden neurons, a window dimension *N* of 5 and with a number of components extracted by the ICA algorithm *n* of 10. In this case the SSE_V was 167 and the SSE_L was 166. The classification results over the prototypes of the learning and the validations sets can be observed in table 1.

Table 1. Classification results over the learning and the validation sets reached by the network with the best generalization.

SET	F	C_L	C_M	C_H	C_V	L	S	SSE
Learning	94.4	91.4	98.3	98.9	96.5	100	100	166
Validation	65.9	85.2	96.2	90.5	93.3	92.4	100	167

In Figure 1 an example of an Iberian Peninsula Meteosat image segmentation can be observed. For the final classification a new class, the indecision class (I), has been added. We consider that one pixel belongs to a class when the output of the neuron representative of this class is bigger than 0.6 and the others outputs are least than 0.4. In other case the pixel is considered to belong to the indecision class.

5 Conclusions

The classification results obtained in table 1 show that ICA is a very efficient feature extraction technique that can be applied at the problem at issue.
At this point we want to compare the classification results obtained in this paper with the results obtained in [13]. In [13] the classification was performed from an initial set of several statistical features based on the gray level co-occurrence matrix (GLCM) proposed by Welch [1]. This initial set of features was made up of 144 parameters and to reduce its dimensionality three methods for feature selection were studied and compared, genetic algorithms (GA), principal component analysis (PCA) and independent component analysis (ICA). To compare the results we use the test set. In table 2 we can observe the classification results obtained over the test set in [13] and the ones obtained in this paper.

Table 2. Classification results over the test set with the methods proposed in [13] and the method proposed in this paper.

	Test set							
Algorithm	F	C_L	C_M	C_H	C_V	L	S	SSE_T
GA	92.7	86.5	77.6	100	75.9	100	100	167
PCAR	84.6	69.7	78.8	97.3	76.8	97.6	100	194
PCANR	62.6	94.2	94.1	91.8	60.7	69.6	99.3	262
ICA (GLCM)	87	76.8	93.5	93.2	81.2	100	100	165
ICA (Images)	**68.3**	**67.3**	**95.9**	**95.9**	**91.1**	**99.2**	**100**	**180**

The results obtained in table 2 show that the method proposed in this paper offers better classification results than the results obtained applying PCA over the 144-dimensional characteristic vector proposed by Welch [1]. But the classification results are worse than the ones obtained applying ICA and GA over the 144-dimensional characteristic vector.
From the meteorology point of view the classification results obtained in this work could be considered the best because this method works quite well over the all classes defined except for the fog (F) and low clouds (C_L) classes. These classes (F and C_L) are very similar and it is very difficult to distinguish between them even for the meteorologists. Initial works concerning this issue did not include the class F and they consider it included into the C_L class.

Acknowledgements.

This study was supported by the Ministerio de Ciencia y Tecnología under project TIC2001-0881.

Thanks are due to EUMETSAT for kindly providing the Meteosat data and to project 1FD970723 (financed by the FEDER and Ministerio de Educación).

References

1. Welch, R.M., Kuo K. S., Sengupta S. K., and Chen D. W.: Cloud field classification based upon high spatial resolution textural feature (I): Gray-level cooccurrence matrix approach. J. Geophys. Res., vol. 93, (oct. 1988) 12633-81.
2. Lee J., Weger R. C., Sengupta S. K. And Welch R.M.: A Neural Network Approach to Cloud Classification. IEEE Transactions on Geoscience and Remote Sensing, vol. 28, no. 5, pp. 846-855, Sept. 1990.
3. M. Macías, F.J. López, A. Serrano and A. Astillero: "A Comparative Study of two Neural Models for Cloud Screening of Iberian Peninsula Meteosat Images", Lecture Notes in Computer Science 2085, Bio-inspired applications of connectionism, pp. 184-191, 2001.
4. A. Astillero, A Serrano, M. Núñez, J.A. García, M. Macías and H.M. Gónzalez: "A Study of the evolution of the cloud cover over Cáceres (Spain) along 1997, estimated from Meteosat images", Proceedings of the 2001 EUMETSAT Meteorological Satellite Data Users' Conference, pp. 353-359, 2001
5. Bankert, R. L et al.,: Cloud Classification of AVHRR Imagery in Maritime Regions Using a Probabilistic Neural Network. Journal of Applied. Meteorology, 33, (1994) 909-918.
6. B. Tian, M. A. Shaikh, M R. Azimi, T. H. Vonder Haar, and D. Reinke, "An study of neural network-based cloud classification using textural and spectral features," IEE trans. Neural Networks, vol. 10, pp. 138-151, 1999.
7. B. Tian, M. R. Azimi, T. H. Vonder Haar, and D. Reinke, "Temporal Updating Scheme for Probabilistic Neural Network with Application to Satellite Cloud Classification," IEEE trans. Neural Networks, Vol. 11, no. 4, pp. 903-918, Jul. 2000.
8. R. M. Welch et al., "Polar cloud and surface classification using AVHRR imagery: An intercomparison of methods," J. Appl. Meteorol., vol. 31, pp. 405-420, May 1992.
9. N. Lamei et al., "Cloud-type discrimitation via multispectral textural analysis," Opt. Eng., vol. 33, pp. 1303-1313, Apr. 1994.
10. R. M. Haralick et al., "Textural features for image classification", IEEE trans. Syst., Man, Cybern., vol. SMC-3, pp. 610-621, Mar. 1973.
11. M. F. Aug.eijin, "Performance evaluation of texture measures for ground cover identification in satellite images by means of a neural.network classifier," IEEE trans. Geosc. Remote Sensing, vol. 33, pp. 616-625, May 1995.
12. M. Macías, C. J. Garcia, H. M. Velasco, R. Gallardo, A. Serrano "A comparison of PCA and GA selected features for cloud field classification", Lectures Notes in Computer Science (Lectures Notes in Artificial Intelligence), vol., 527, pp., 42-49, 2002.
13. C.J. García, M. Macías, A. Serrano, H.M. González and R. Gallardo, "A comparison of PCA, ICA and GA selected features for cloud field classification", Journal of Intelligent & Fuzzy Systems. Accepted for publication.
14. P. Comon. Independent component analysis – a new concept? Signal Processing, 36: 287-314, 1994.
15. Hyvärinen A., Karhunen J., Oja E. Independent Component Analysis. John Wiley and Sons, 2001.
16. A. Hyvärinen, E. Oja, P. Hover and J. Hurri. Image feature extraction by sparse coding and independent component analisys. In Proc. Int. Conf. On Pattern Recognition (ICPR'98), pp.1268-1273, Brisbane, Australia, 1998.
17. C. Jutten and J. Herault, "Blind separation of sources, part I: an adaptive algorithm based on neuromimetic architecture" Signal Processing, vol.24, no. 1, pp. 1-10, july 1991.
18. J. Hérault and C. Jutten, Réseaux neuronaux et traitement du signal, Hermes, 1994.

19. J. Karhunen and J. Joutsensalo, “Generalizations of principal component analysis, optimization problems, and neural networks” Neural Networks, vol. 8, no. 4, pp. 549--562, 1995
20. A. Cichocki, R. Unbehauen and E. Rummert, “Robust learning algorithm for blind separation of signals”, Electronics Letters, vol.30, No.17, 18th August 1994, pp.1386-1387.
21. P. Comon, “Independent component analysis, a new concept?,” Signal Processing, vol. 36, no. 3, pp. 287--314, April 1994.
22. A. Hyvärinen and E. Oja. Independent Component Analysis: Algorithms and Applications. Neural Networks, 13(4-5):411-430, 2000.
23. A. Hyvärinen .Fast and robust fixed-point algorithms for independent component analysis. IEEE Trans. On Neural Networks, 10(3): 626-634, 1999.
24. R and C code implementation of the fastICA package. http://www.cis.hut.fi/projects/ica/fastica/.
25. The R project for statistical computing. http://www.r-project.org/
26. M. Riedmiller, M., Braun, L.: A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. In Proceedings of the IEEE International Conference on Neural Networks 1993 (ICNN 93), 1993.

Neural Solutions for High Range Resolution Radar Classification

R. Gil-Pita, P. Jarabo-Amores, R. Vicen-Bueno and M. Rosa-Zurera

Departamento de Teoría de la Señal y Comunicaciones
Escuela Politécnica, Universidad de Alcalá
Ctra. Madrid-Barcelona, km. 33.600, 28871, Alcalá de Henares - Madrid (SPAIN)
E-mails:{roberto.gil,mpilar.jarabo,raul.vicen,manuel.rosa}@uah.es

Abstract. In this paper the application of neural networks to Automatic Target Recognition (ATR) using a High Range Resolution radar is studied. Both Multi-layer Perceptrons (MLP) and Radial Basis Function Networks (RBFN) have been used. RBFNs can achieve very good results with a considerably small size of the training set, but they require a high number of radial basis functions to implement the classifier rule. MLPs need a high number of training patterns to achieve good results but when the training set size is higher enough, the performance of the MLP-based classifier approaches the results obtained with RBFNs, but with lower computational complexity. Taking into consideration the complexity of the HRR radar data, the choice between these two kind of neural networks is not easy. The computational capability and the available data set size should be considered in order to choose the best architecture. MLPs must be considered when a low computational complexity is required, and when a large training set is available; RBFNs must be used when the computational complexity is not constrained, or when only few data patterns are available.

1 Introduction

High resolution radar (HRR) generates data sets that have significantly different properties from others used in automatic target recognition (ATR) [1]. This kind of radar uses broad-band linear frequency modulation or step frequency waveforms to measure range profiles (signatures) of targets [2]. Target signatures are sensitive to minor changes in azimuth, elevation and time, because the scattering into a range cell comes from the combination of scatter contributions of different points in that cell.

In this paper, different possibilities are evaluated in order to solve a classification problem based on this kind of signals. In the literature this is called an "Automatic Target Recognition" (ATR) problem. We deal with the usage of different Neural Network based solutions [3].

Automatic Target Recognition (ATR) can be formulated as a multiple hypothesis test. The goal is to classify an example which has not been seen before, based on knowledge extracted from previous examples. The probability density

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 559-566, 2003.

functions (pdfs) of the classes usually are unknown, and only a finite set of solved cases (training set) is available. Different techniques have been developed, which have different trade-offs in terms of accuracy and complexity, and make different assumptions about the problem and its solution.

The classical approach to HRR radar target identification uses the entire range profile as feature vector. One of the most popular choices is a constrained quadratic classifier [4]. Other strategies are based on extracting reliable features from radar profiles [5][6][7]. On the other hand, neural networks (NNs) have been successfully applied to pattern recognition [8]. An overview of neural networks application to ATR can be found in [9]. Modern approaches try to combine neural networks with tuned pre-processing methods to enhance the performance of ATR systems [10].

2 Problem Formulation

Our objective is to classify patterns that belong to six different classes of HRR radar profiles corresponding to six kind of aircrafts. For each class 1,071 radar profiles are available. The length of each profile is 128. The pose of the target is head-on with an azimuth range of $\pm 25^o$ and elevations of -20^o to 0^o in one degree increments. For each experiment, this data set is divided in three sets: a training set, that is composed of M randomly selected signals from the data set; a validation set, composed of other M signals, used during training to estimate the probability of classification error, and a test set composed of the remaining signals, used to measure the accuracy of classification after training. In this paper experiments with $M = 60, 240, 420$ and 600 are used. So, the relation between the results obtained with each classification method and data availability is studied. Figure 1 shows range profiles of two targets, illustrating the extreme variability in the radar signature with minor changes in azimuth, elevation and time.

To specify the performance of the classifier, the *probability of correct classification* (Pcc), the *probability of misclassification* (Pmc) or the *error rate* can be used. The probability of correct classification is the probability of classifying correctly a target. The probability of misclassification is the probability of classifying wrongly a target ($Pmc = 1 - Pcc$). Finally, the error rate expresses the percentage of classification errors when a set of known targets is presented to the system.

3 ATR with classification methods

In this section, Neural Networks (NN) are proposed as a solution to ATR, due to their ability to learn from their environment, and to improve performance in some sense through learning. NN can approximate the Bayessian optimum detector [11] and it has been demonstrated [12],[8] that the back-propagation algorithm applied to a simple feed-forward network approximates the optimum Bayessian classifier, when using the mean square error criterion.

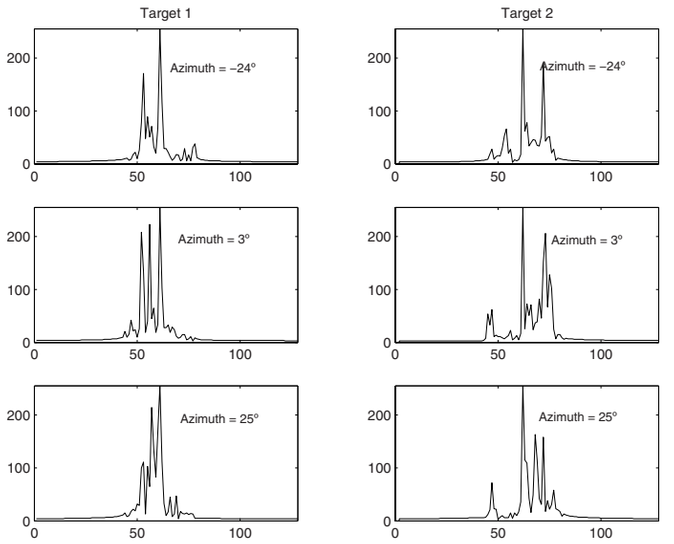


Fig. 1. Radar signatures of two different targets for an elevation angle of -20^o and different azimuth angles

For comparison purposes, results using the k-nearest neighbor classifier [13] are also included.

3.1 Classification method based on a Multi-Layer Perceptron

Perceptron was developed by F. Rosenblatt [14] in the 1950s for optical character recognition. The perceptron has multiple inputs fully connected to an output layer with multiple outputs. Each output y_j can be obtained by a linear combination of the inputs, applied to a non linear function called activation function. In this work we are going to considerer the logistic function given in 1 as the activation function.

$$L(x) = \frac{1}{1 + \exp(-x)} \tag{1}$$

Multilayer perceptrons (MLPs) extend the perceptron by cascading one or more extra layer of processing elements. These extra layers are called hidden layers, since their elements are not connected directly to the external world. Figure 2 shows a one-hidden-layer MLP with K inputs, N neurons in the hidden layer and 6 outputs.

On the other hand, with the data described in section 2, Cybenko [15] states that a single hidden layer is sufficient for a MLP to compute an uniform approximation to the conditional probabilities, from a given training set and the

corresponding desired target outputs. So, a MLP with a hidden layer has been considered. The activation function of every neuron in the hidden layer is the logistic one. In order to study the influence of the training set size, the experiment has been repeated for the four designed partition of the data. These partitions are referenced according to the training set size ($M = 60, 240, 420$ and 600).

According to this, the size of the MLP is $K = 40$ input nodes where the central samples of each pattern are applied, and 6 outputs units, one for each class. The number of neurons N on the hidden layer is an input parameter in the experiment.

Table 1. Best average error rates (%) for the MLP with N hidden neurons trained with M patterns

	N=4	*N=8*	*N=12*	*N=16*	*N=20*	*N=24*	*N=28*	*N=32*	*N=36*	*N=40*
M=60	42.10	44.91	34.55	27.64	16.33	17.05	25.45	15.39	19.63	16.33
M=240	32.19	12.55	15.01	12.48	14.99	14.97	12.80	13.77	12.27	9.26
M=420	14.54	12.00	10.59	10.82	13.76	10.62	11.88	8.21	11.00	10.82
M=600	12.56	8.25	8.58	11.29	9.19	7.26	6.65	6.32	5.91	6.76

Moreover, each experiment has been repeated 10 times, and the best network has been selected using the validation sets. The MLP has been trained using the Levenberg-Marquardt algorithm [16], with the four different training sets, and the respective validation sets. The error rate classifying the test samples for the best training session of each value of M and N are presented in table 1.

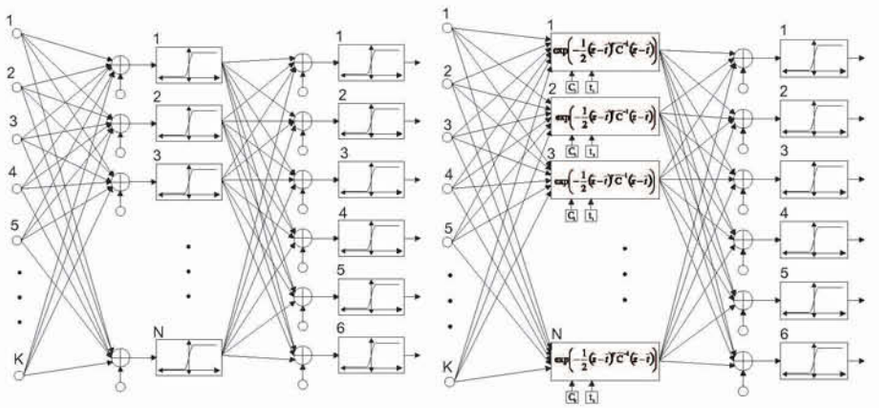


Fig. 2. MLP and RBFN architectures

The best results for each M value are now compared with those obtained using the k-nearest neighbor classifier. For comparison purposes, the error rate and the computational complexity, measured in number of simple operations, are used. Results are presented in tables 2 and 3.

As can be observed, the use of MLPs involve an increase in the error rate, but highly improves the computational cost. Moreover, MLP computational cost does not depend on set size, so MLPs are advisable when large training sets are available.

Table 2. Number of operations and average error rates (%) for the MLP trained with M samples and with best N value applied to the test sets

	M=60	*M=240*	*M=420*	*M=600*
hidden neurons (N)	32	40	32	36
error rate	15.39	9.26	8.21	5.91
operations	3046	3806	3046	3426

Table 3. Error rates (%), best k value and number of simple operations for the k-Nearest Neighbor method with the test sets

	M=60	*M=240*	*M=420*	*M=600*
best k value	1	4	4	9
error rate	14.69	10.15	7.43	6.65
operations	7261	29758	52078	77373

3.2 Classification method based on a Radial Basis Function Network

In Radial-Basis Function Networks, the function associated to the hidden units (*radial-basis function*) is usually the multivariate normal function. For the $i-th$ hidden unit it can be expressed using (2).

$$G_i(\mathbf{z}) = \frac{|\mathbf{C_i}|^{\frac{-1}{2}}}{(2\pi)^{\frac{n}{2}}} exp\{\frac{-(\mathbf{z}-\mathbf{t_i})^T \mathbf{C_i}^{-1}(\mathbf{z}-\mathbf{t_i})}{2}\} \tag{2}$$

where C_i is the covariance matrix, which controls the smoothness properties of the function.

Using the weighted norm [17] defined in (3), equation (2) can be expressed in (4).

$$\|\mathbf{z}\|_C^2 = \frac{1}{2}\mathbf{z}^T\mathbf{C}^{-1}\mathbf{z} \tag{3}$$

$$G_i(\mathbf{z}) = \frac{|\mathbf{C_i}|^{\frac{-1}{2}}}{(2\pi)^{\frac{n}{2}}} \exp(-\|\mathbf{z} - \mathbf{t_i}\|^2_{C_i}) \tag{4}$$

Taking into consideration that the hidden unit output is multiplied by a weight that will be adjusted during the training, the considered RBF function in figure 2 is:

$$G(z) = \exp(-\|\mathbf{z} - \mathbf{t_i}\|^2_{\mathbf{C_i}}) \tag{5}$$

The matrices $\mathbf{C_i}$ can be set to a scalar multiple of the unit matrix, to a diagonal matrix with different diagonal elements or to a non-diagonal matrix. In this case we have set them to a scalar multiple of the unit matrix.

For training the RBFNs, we have applied a three learning phases strategy [17][18][19]. The centres of the radial basis functions are determined by fitting a gaussian mixture model with circular covariances using the EM algorithm. The mixture model is initialized using a small number of iterations of the k-means algorithm; the basis function widths are set to the maximum inter-centre squared distance; and the hidden to output weights that give rise to the least squares solution can be determined using the LMS algorithm.

The experiment has been repeated for each partition and for different number of hidden units N. This value has been chosen considering the training set size M. So, for a given M, values from $N = 0.05M$ to $N = 0.5M$ in steps of $0.05M$ have been selected. For each training, the validation set has been used to control the training. Moreover, the training has been repeated 10 times and the best case using the validation set has been selected. Table 4 shows results obtained with the test set, depending on the training set size M and the normalized hidden layer size $n = N/M$.

Table 4. Average error rates (%) for the RBFN trained with M samples and N = n*M hidden neurons

M	n=0.05	n=0.10	n=0.15	n=0.20	n=0.25	n=0.30	n=0.35	n=0.40	n=0.45	n=0.50
60	48.82	22.03	15.10	13.46	10.36	13.40	15.73	11.72	12.74	16.80
240	11.97	9.05	7.60	7.30	6.59	6.82	6.26	6.19	6.65	6.98
420	7.86	6.55	4.47	4.54	4.57	4.42	4.77	4.59	4.64	5.10
600	6.46	4.66	3.86	3.15	3.34	3.53	3.45	4.06	4.33	4.47

Once RBFNs are trained, validation sets are used to select the best trained network for each M. The number of simple operations, the error rate estimated with the test set and the number of hidden neurons are shown in table 5.

Compared with the ones obtained with MLP and kNN methods, results obtained show that the error rates with the RBFN based classifier are significantly lower than those obtained with the MLP or the kNN ones. On the other hand,

the computational complexity is lower than the k-nearest neighbor one, but high compared with the computational complexity of the MLP based classifier. Another characteristic of RBFN is the dependence of the computational cost on the training set size. Due to the complexity of the HRR data, the number of hidden layer neurons directly depends on the training set size. In order to compare the results obtained in previous sections, we summarize them in table 6.

Table 5. Number of operations and average error rates (%) for the RBFN trained with M samples and best N value applied to the test sets

	M=60	*M=240*	*M=420*	*M=600*
best N	15	96	126	120
error rate	10.36	6.19	4.42	3.15
operations	2661	16998	22308	21246

Table 6. Average error rates and computational complexity (%/number of operations) for the studied methods applied to the test sets

error/cost	*M=60*	*M=240*	*M=420*	*M=600*
kNN	14.69 / 7261	10.15 / 29758	7.43 / 52078	6.65 / 77373
MLP	15.39 / 3046	9.26 / 3806	8.21 / 3046	5.91 / 3426
RBFN	10.36 / 2661	6.19 / 16988	4.42 / 22308	3.15 / 21246

4 Conclusions

In this paper the application of neural networks to Automatic Target Recognition (ATR) using a High Range Resolution radar is studied. Due to the complexity of High Range Resolution radar data, the classification objective is a difficult task.

Both MLPs and RBFNs have been trained using different network structure and training set sizes. Multilayer Perceptrons (MLP) need a high number of training patterns to achieve good results. When the training set size is higher enough, the performance of the MLP-based classifier approaches the results obtained with the k-nearest neighbor, but with lower computational complexity. On the other hand, Radial Basis Function Networks (RBFN) can achieve very good results with a considerably small size of the training set, but they require a high number of radial basis functions to implement the classifier rule.

Taking into consideration the complexity of the HRR radar data, the choice between these two kind of neural networks is not easy. The computational capability and the available data set size should be considered in order to choose

the best architecture. MLPs must be considered when a low computational complexity is required, and when a large training set is available; RBFNs must be used when the computational complexity is not constrained, or when only few data patterns are available.

References

1. S. P. Jacobs, J. A. O'Sullivan, "Automatic Target Recognition Using Sequences of High Resolution Radar Range-Profiles", *IEEE Trans. on Aerospace and Electronic Systems*, vol. 36, no. 2, April, 2000.
2. C.R. Smith, P.M. Coggans, "Radar Target Identification", *IEEE Antennas and Propagation Magazine*, vol. 35, no. 2, pp. 27-37, April 1993.
3. M. W. Roth, "Survey of Neural Network Technology for Automatic Target Recognition", *IEEE Trans. on Neural Networks*, vol. 1, no. 1, March, 1990.
4. K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Boston, Academic Press, 1990.
5. R.A. Mitchell, et al., "Robust Statistical Feature Based Aircraft Identification", *IEEE Trans. on Aerospace and Electronic Systems*, vol. 35, no. 3, 1077 -1094, 1999.
6. D.E. Nelson, J.A. Starzyk, "Advance Feature Selection Methodology for Automatic Target Recognition", *Proceedings of the* 29^{th} *Southeastern Symposium on System Theory*, 1997, pp. 24-28.
7. D.E. Nelson, J.A. Starzyk, "High Range Resolution Radar Signal Classification. A Partitioned Rough Set Approach." *Proceedings of the* 33^{rd} *Southeastern Symposium on System Theory*, 2001, pp. 21-24.
8. E.A. Wan, "Neural Network Classification: A Bayesian Interpretation", *IEEE Trans. on Neural Networks*, vol. 1, no. 4, pp. 303-305, December 1990.
9. S.K. Rogers et al., "Neural Networks for Automatic Target Recognition", *Neural Networks*, vol. 8, no. 7/8, pp. 1153-1184, April 2000.
10. J.K. Aggarwal et al. "A Comparative Study of Three Paradigms for Object Recognition - Bayesian Statistics, Neural Networks and Expert Systems", *Advances in Image Understanding: A Festschrift for A. R.*, Computer Society Press, 1996.
11. J. W. Watterson, "An Optimun Multilayer Perceptron Neural Receiver for Signal Detection", *IEEE Trans. on Neural Networks*, vol. 1, no. 4, December, 1990.
12. D.W. Ruck, et al., "The Multilayer Perceptron as an Aproximation to a Bayes Optimal Discriminant Function", *IEEE Trans. on Neural Networks*, vol. 1, no. 4, pp. 296-298, April 1990.
13. A. A. Kisienski, et al., "Low-frequency approach to target identification", *Proceedings of the IEEE*, 63, pp. 1651-1659, 1975.
14. F. Rosenblatt, *Principles of Neurodynamics*, Spartan books, New York, 1962.
15. G. Cybenko, "Approximation by superpositions of a sigmoidal function", *Mathematics of Control, Signals and Systems*, vol. 2, pp. 303-314, 1989.
16. M.T. Hagan, M.B. Menhaj, "Training Feedforward Networks with the Marquardt Algorithm", *IEEE Trans. on Neural Networks*, vol. 5, no. 6, pp. 989-993, November 1994.
17. S. Haykin, *Neural networks. A comprehensive foundation (second edition)*, Prentice-Hall Inc., 1999.
18. C.M. Bishop, *Neural networks for pattern recognition*, Oxford University Press Inc., 1995.
19. F. Schwenker, H.A. Kestler, G. Palm, "Three learning phases for radial-basis-function networks", *Neural Networks*, Vol. 14, Issue 4-5, 2001, pp. 439-458.

On the application of Associative Morphological Memories to Hyperspectral Image Analysis

Manuel Graña *, Josune Gallego, F. Javier Torrealdea, and Alicia d'Anjou

Dept. CCIA UPV/EHU
ccpgrrom@si.ehu.es

Abstract. We propose a spectrum selection procedure from hyperspectral images, which uses the Autoassociative Morphological Memories (AMM) as detectors of morphological independence conditions. Selected spectra may be used as endmembers for spectral unmixing. Endmember spectra lie in the vertices of a convex region that covers the image pixel spectra. Therefore, morphological independence is a necessary condition for these vertices. The selective sensitivity of AMM's to erosive and dilative noise allows their use as morphological independence detectors.

1 Introduction

Hyperspectral sensing in hundreds of spectral bands allows the recognition of physical materials in image pixels, however as these pixels are frequently a combination of materials, a frequently needed image analysis procedure is to decompose each pixel spectrum into their constituent material spectra, a process known as "spectral unmixing" [4]. In this paper we assume the linear mixture model. We present a computational procedure for the detection in hyperspectral images of pixel spectra that may serve as endmembers for spectral unmixing. This procedure profits from the selective sensitivity to noise of the Associative Morphological Memories for the detection of the morphological independence conditions that are a necessary condition of endmember spectra. Although we have successfully tested the procedure on the original non-negative radiance data, it works better after the original spectral data is shifted to zero mean. In its actual implementation it works in a single pass over the image, which is a desirable feature given the large computational cost of processing hyperspectral images. The procedure is unsupervised and does not need the explicit setting of the number of endmembers searched for. For a qualitative evaluation of the results we present the abundance images obtained from the well-known Indian Pines benchmark image [10] with the endmembers obtained with our procedure and the CCA method [3].

In short, Morphological Neural Networks are those that somehow involve the maximum and/or minimum (supremum and/or infimum) operators. The Associative Morphological Memories [6], [7], [8] are the morphological counterpart

* The authors received partial support of the Ministerio de Ciencia y Tecnologia under grant TIC2000-0739-C04-02

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 567-574, 2003.

of the well known Hopfield Associative Memories [2]. AMM's are constructed as correlation matrices computed by either Min or Max matrix product. Dual constructions can be made using the dual Min and Max operators. The AMM are selectively sensitive to specific types of noise (erosive and dilative noise). The notion of morphological independence and morphological strong independence was introduced in [8] to study the construction of robust to general noise AMM's. It was established that AMM's are able to robustly store and recall morphologically strongly independent sets of patterns. Applying to the AMM as input for recalling a pattern morphologically dependent on one of the stored patterns gives the stored pattern. When the input pattern is morphologically independent of the stored patterns, the result of recall is a morphological polynomial on the stored patterns [9]. In essence our procedure tests if the recalled pattern is different to the stored patterns to detect morphologically independent patterns.

The structure of the paper is as follows: In section 2 we review the definition of the linear mixing model and the spectral unmixing procedure. In section 3 we review the basic definitions and properties of AMM's. Section 4 gives our algorithm of endmember selection for remote sensing hyperspectral images. Section 5 presents some experimental results of the proposed algorithm. Section 6 gives our conclusions and directions of future work.

2 Spectral unmixing and the linear mixing model

The linear mixing model [4] can be expressed as follows:

$$\mathbf{x} = \sum_{i=1}^{M} a_i \mathbf{s}_i + \mathbf{w} = \mathbf{S}\mathbf{a} + \mathbf{w}, \tag{1}$$

where $\mathbf{x}$ is the d-dimension received pixel spectrum vector, $\mathbf{S}$ is the $d \times M$ matrix whose columns are the d-dimension endmembers $\mathbf{s}_i, i = 1, .., M$, $\mathbf{a}$ is the M-dimension fractional abundance vector, and $\mathbf{w}$ is the d-dimension additive observation noise vector. The linear mixing model is subjected to two constraints on the abundance coefficients. First, to be physically meaningful, all abundance coefficients must be non-negative $a_i \geq 0, i = 1, .., M$. Second, to account for the entire composition, they must be fully additive $\sum_{i=1}^{M} a_i = 1$.

The task of selecting image pixel spectra that may be used as endmember is the focus of this paper. The first significant work on the automated induction of endmember spectra from image data is [1], whose starting observation is that the scatter plots of remotely sensed data are tear shaped or pyramidal, if two or three spectral bands are considered. The apex lies in the so-called dark point. The endmember detection becomes the search for non-orthogonal planes that enclose the data defining a minimum volume simplex, hence the name of the method. The method is computationally expensive and requires the prior specification of the number of endmenbers. A recent approach to the automatic endmember detection is the Convex Cone Analysis (CCA) method proposed in [3], applied to

target detection. The CCA selects the greatest eigenvalue eigenvectors, as many as the specified number of endmembers. These eigenvectors define the basis of the convex cone that covers the image data. The vertices of the convex cone correspond to spectra with as many zero elements as the number of eigenvectors minus one. The search for the convex cone vertices involves the exploration of the combination of bands and the solution of a linear system for each combination. The complexity of the search for these vertices is $O\left(b^c\right)$ where b is the number of bands and c the number of eigenvectors. At present we use a raw random search to obtain the experimental results reported below. Another approach is the modelling by Markov Random Fields and the detection of spatially consistent regions whose spectra will be assumed as endmembers [5].

Once the endmember spectra have been determined the spectral unmixing is the computation of the matrix inversion that gives the fractional abundance of each endmember in each pixel spectra. The simplest approach is the unconstrained least squared error estimation given by:

$$\widehat{\mathbf{a}} = \left(\mathbf{S}^T\mathbf{S}\right)^{-1}\mathbf{S}^T\mathbf{x}. \tag{2}$$

The abundance coefficients that result from this computation do not necessarily fulfill the non-negativity and full additivity conditions. It is possible to enforce each condition separately, but rather difficult to enforce both simultaneously [4]. As our aim is to test an endmember determination procedure, we will use unconstrained estimation (2) to compute the abundance images. We will scale and shift the abundance images to present them as grayscale images. This manipulation is intended to enhance the visualization but introduces some deformation of the relative values of the abundance coefficients for the same pixel.

3 Associative Morphological Memories

The work on Associative Morphological Memories stems from the consideration of an algebraic lattice structure $(\mathbb{R}, \vee, \wedge, +)$ as the alternative to the algebraic $(\mathbb{R}, +, \cdot)$ framework for the definition of Neural Networks computation [6] [7]. The operators $\vee$ and $\wedge$ denote, respectively, the discrete max and min operators (resp. sup and inf in a continuous setting), which correspond to the morphological dilation and erosion operators, respectively. Given a set of input/output pairs of pattern $(X, Y) = \left\{\left(\mathbf{x}^\xi, \mathbf{y}^\xi\right); \xi = 1, .., k\right\}$, an heteroassociative neural network based on the pattern's crosscorrelation [2] is built up as $W = \sum_\xi \mathbf{y}^\xi \cdot \left(\mathbf{x}^\xi\right)'$. Mimicking this construction procedure [6], [7] propose the following constructions of Heteroassociative Morphological Memories (HMM's):

$$W_{XY} = \bigwedge_{\xi=1}^{k}\left[\mathbf{y}^\xi \times \left(-\mathbf{x}^\xi\right)'\right] \text{ and } M_{XY} = \bigvee_{\xi=1}^{k}\left[\mathbf{y}^\xi \times \left(-\mathbf{x}^\xi\right)'\right], \tag{3}$$

where $\times$ is any of the $\boxtimes$ or $\boxtimes$ operators. Here $\boxtimes$ and $\boxtimes$ denote the max and min matrix product, respectively defined as follows:

$$C = A \boxtimes B = [c_{ij}] \Leftrightarrow c_{ij} = \bigvee_{k=1..n}\left\{a_{ik} + b_{kj}\right\}, \tag{4}$$

$$C = A \boxtimes B = [c_{ij}] \Leftrightarrow c_{ij} = \bigwedge_{k=1..n} \{a_{ik} + b_{kj}\}. \quad (5)$$

If $X = Y$ then the HMM memories are Autoassociative Morphological Memories (AMM). Conditions of perfect recall by the HMM's and AMM's of the stored patterns are proved in [6],[7]. The AMM's are able to store and recall any set of patterns: $W_{XX} \boxtimes X = X = M_{XX} \boxtimes X$, for any X.

These results hold when we try to recover the output patterns from the noise-free input pattern. Let it be $\widetilde{\mathbf{x}}^\gamma$ a noisy version of $\mathbf{x}^\gamma$. If $\widetilde{\mathbf{x}}^\gamma \leq \mathbf{x}^\gamma$ then $\widetilde{\mathbf{x}}^\gamma$ is an eroded version of $\mathbf{x}^\gamma$, alternatively we say that $\widetilde{\mathbf{x}}^\gamma$ is subjected to erosive noise. If $\widetilde{\mathbf{x}}^\gamma \geq \mathbf{x}^\gamma$ then $\widetilde{\mathbf{x}}^\gamma$ is a dilated version of $\mathbf{x}^\gamma$, alternatively we say that $\widetilde{\mathbf{x}}^\gamma$ is subjected to dilative noise. Morphological memories are selectively sensitive to these kinds of noise. The conditions of *robust* perfect recall are proven in [6], [7]. Given patterns X, the equality $W_{XX} \boxtimes \widetilde{\mathbf{x}}^\gamma = \mathbf{x}^\gamma$ holds when the noise affecting the pattern is erosive $\widetilde{\mathbf{x}}^\gamma \leq \mathbf{x}^\gamma$ and the following relation holds $\forall i \exists j_i; \widetilde{x}^\gamma_{j_i} = x^\gamma_{j_i} \vee \left(\bigvee_{\xi \neq \gamma} \left(x^\gamma_i - x^\xi_i + x^\xi_{j_i} \right) \right)$. Similarly, the equality $M_{XY} \boxtimes \widetilde{\mathbf{x}}^\gamma = \mathbf{x}^\gamma$ holds when the noise affecting the pattern is dilative $\widetilde{\mathbf{x}}^\gamma \geq \mathbf{x}^\gamma$ and the following relation holds: $\forall i \exists j_i; \widetilde{x}^\gamma_{j_i} = x^\gamma_{j_i} \wedge \left(\bigwedge_{\xi \neq \gamma} \left(x^\gamma_i - x^\xi_i + x^\xi_{j_i} \right) \right)$. Therefore, the AMM will fail to recall the pattern if the noise is a mixture of erosive and dilative noise.

To obtain general noise robustness [6], [8], [9] proposed the kernel method. Related to the construction of the kernels, [8] introduced the notion of morphological independence. Here we distinguish erosive and dilative versions of this definition: Given a set of pattern vectors $X = \left(\mathbf{x}^1, ..., \mathbf{x}^k\right)$, a pattern vector $\mathbf{y}$ is said to be morphologically independent of X in the erosive sense if $\mathbf{y} \not\leq \mathbf{x}^\gamma; \gamma = \{1, .., k\}$, and morphologically independent of X in the dilative sense if $\mathbf{y} \not\geq \mathbf{x}^\gamma; \gamma = \{1, .., k\}$. The set of pattern vectors X is said to be morphologically independent in either sense when all the patterns are morphologically independent of the remaining patterns in the set. For the current application we want to use AMM as detectors of the set extreme points, to obtain a rough approximation of the minimal simplex that covers the data points. We note that given a set of pattern vectors $X = \left(\mathbf{x}^1, ..., \mathbf{x}^k\right)$, and the erosive W_{XX} and dilative M_{XX} memories constructed from it, and a test pattern $\mathbf{y} \notin X$, if $\mathbf{y}$ is morphologically independent of X in the erosive sense, then $W_{XX} \boxtimes \mathbf{y} \notin X$. Also, if $\mathbf{y}$ is morphologically independent of X in the dilative sense, then $M_{XX} \boxtimes \mathbf{y} \notin X$. Therefore the AMM's can be used as detectors of morphological independence.

The endmembers that we are searching for define a high dimensional box that covers the image spectral data. Although, for convenience, we will perform a shift of the data to the space origin, the endmember spectrum will be taken from the original image. They are morphologically independent vectors both in the erosive and dilative senses, and they enclose the remaining vectors. Let us consider integer-valued vectors. Given a set of pattern vectors $X = \left(\mathbf{x}^1, ..., \mathbf{x}^k\right)$ and the erosive W_{XX} and dilative M_{XX} memories constructed from it, if a test pattern $\mathbf{y} < \mathbf{x}^\gamma$ for some $\gamma \in \{1, .., k\}$ then $W_{XX} \boxtimes \mathbf{y} \notin X$. Also, if the test pattern $\mathbf{y} > \mathbf{x}^\gamma$ for some $\gamma \in \{1, .., k\}$ then $M_{XX} \boxtimes \mathbf{y} \geq \notin X$. Therefore, the AMM will be useless for the detection of morphologically independent integer

valued patterns. However, if we consider the binary vectors obtained as the sign of the vector components, then morphological independence may be detected as suggested above: The already detected endmembers are used to build the erosive and dilative AMM. If the output recalled by a new pattern does not coincide with any of the endmembers, then the new pattern is a new endmember.

4 The selection of endmember spectra

The endmembers of a given hyperspectral image correspond to the vertices of the minimal simplex that covers the data points. The region of the space defined by a set of vectors morphologically independent in both erosive and dilative senses is a high dimensional box that approaches the minimal simplex covering the data points. Let us define $\left\{\mathbf{f}\left(i,j\right) \in \mathbb{R}^d; i = 1,..,n; j = 1,..,m\right\}$ the d-band hyperspectral image, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ the vectors of the mean and standard deviations of each band computed over the image, α the noise correction factor and E the set of endmembers discovered. The noise amplitude in (1) is estimated as $\boldsymbol{\sigma}$, the patterns are corrected by the addition and substraction of $\alpha\boldsymbol{\sigma}$, before being presented to the AMM's. The confidence level α controls the amount of flexibility in the discovering of new endmembers. Let us denote by the expression $\mathbf{x} > \mathbf{0}$ the construction of the binary vector $\left(\left\{b_i = 1 \text{ if } x_i > 0; b_i = 0 \text{ if } x_i \leq 0\right\}; i = 1,..,n\right)$.

The steps in the procedure are the following (Note that the endmember spectra are the pixel spectra from the original image):

1. Compute the zero mean image
   $\{\mathbf{f}^c\left(i,j\right) = \mathbf{f}\left(i,j\right) - \boldsymbol{\mu}; i = 1,..,n; j = 1,..,m\}$.
2. Initialize the set of endmembers $E = \{\mathbf{e}_1\}$ with a pixel spectra randomly picked from the image. Initialize the set of morphologically independent binary signatures $X = \{\mathbf{x}_1\} = \left\{\left(e_k^1 > 0; k = 1,..,d\right)\right\}$
3. Construct the AMM's based on the morphologically independent binary signatures: M_{XX} and W_{XX}.
4. For each pixel $\mathbf{f}^c\left(i,j\right)$
   (a) compute the vector of the signs of the Gaussian noise corrections $\mathbf{f}^+\left(i,j\right) = \left(\mathbf{f}^c\left(i,j\right) + \alpha\boldsymbol{\sigma} > \mathbf{0}\right)$ and $\mathbf{f}^-\left(i,j\right) = \left(\mathbf{f}^c\left(i,j\right) - \alpha\boldsymbol{\sigma} > \mathbf{0}\right)$
   (b) compute the response of the dilative memory $y^+ = M_{XX} \boxtimes \mathbf{f}^+\left(i,j\right)$
   (c) compute the response of the erosive memory$y^- = W_{XX} \boxtimes \mathbf{f}^-\left(i,j\right)$
   (d) if $y^+ \notin X$ and $y^- \notin X$ then $\mathbf{f}^c\left(i,j\right)$ is a new endmember to be added to E, go to step 3 and resume the exploration of the image.
   (e) if $y^+ \in X$ and $\mathbf{f}^c\left(i,j\right) > \mathbf{e}_{y^+}$ the pixel spectral signature is more extreme than the stored endmember, then substitute $\mathbf{e}_{y^+}$ with $\mathbf{f}^c\left(i,j\right)$.
   (f) if $y^- \in X$ and $\mathbf{f}^c\left(i,j\right) < \mathbf{e}_{y^-}$ the pixel is more extreme than the stored endmember, then substitute $\mathbf{e}_{y^-}$ with $\mathbf{f}^c\left(i,j\right)$.
5. The final set of endmembers is the set of original spectral signatures $\mathbf{f}\left(i,j\right)$ of the pixels selected as members of E.

Fig. 1. Ground truth of the Indian Pines image.

5 Experimental results

The hperspectral image used for illustration in this work correspond to the Indian Pines image obtained in 1992 by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) developed by NASA JPL which has 224 contiguous spectral channels covering a spectral region form 0.4 to 2.5 mm in 10 nm steps. It is a 145 by 145 pixel image with 220 spectral bands that contains a distribution of two-thirds of agricultural land and one-third of forest and other elements (two highways, a railway line and some houses and smaller roads). The available ground truth for this image designates 16 mutually exclusive classes of land cover. Figure 1 shows the ground truth as given in [10]. Comparison of the results of careful supervised classification [10] with the abundance images in figure 2 resulting from the detected endmembers, confirm the validity of the abundance images and explain their discrepancies relative to the ground truth areas.

We have applied the procedure described in the previous section to the raw data of the benchmark hyperspectral image, obtaining 8 endmember spectra. The number of endmembers found depends on the initial choice of endmember and on the control parameter α, which was empirically set to $\alpha = 2$. The abundance images resulting from spectral unmixing with these spectra show some regions that match the ground truth, while some new regions appear in the region labeled as background in the ground truth. The comparison with the abundance images obtained from the spectral unmixing with the CCA endmembers contributes also to the qualitative validation of our procedure. This qualitative validation is useful because there is no clear quantitative alternative to validate the approach. The abundance images that result from our procedure are presented in figure 2. As said before, we have also applied the CCA method with the number of endmembers set to 6 (a number of endmembers greater than that gave very large computation times). The abundance images are presented in figure 3. Note that these abundance images are in fact two images repeated three times. The CCA obtains two very different endmember spectra, the remaining four are scaled versions of these two basic spectra. Lack of space prevents us presenting the plot of the endmembers that demonstrates this fact.

Consider, for example, the steel towers identified in the ground truth. The endmember spectra corresponding to the abundance images #5 and #1 in figure 2 may be considered as good steel detector. Curiously enough, the roads are also

clearly drawn in abundance image #1. Image #5 of figures 2 and 3 shows good detection of cultivated land and negative abundance response in woods areas, besides the detection of the steel tower. Our interpretation is that the endmember spectra corresponding to these images is a generic opposite to the vegetal cover spectra. Furthermore, if we consider the fact that due to the early growth stages most of the surface area corresponding to a pixel in the cultivated land is bare soil, we may assume that the endmember that generates this abundance image corresponds to soil cover. On the other hand, abundance image #7 in figure 2 and abundance image #3 in figure 3 clearly detach woods and tree canopy areas. The detection in these images agrees with the ground truth in 1. Besides, the ground truth background class pixels identified also with woods in some areas of these abundance images agree with the results of careful supervised classification experiments reported in [10].

6 Conclusions and Further Work

We have proposed a procedure for the selection of pixel spectra in hyperspectral images that applies the specific noise sensitivity of the Autoassociative Morphological Memories (AMM). The selected pixel spectra may be used as endmembers for spectral unmixing. The procedure does not need the a priori setting of the number of endmembers. Its flexibility in the discovering of endmembers is controlled by the amount of noise correction introduced in the pixel spectral signature. Experimental results demonstrate that the procedure gives a sensible number of endmembers with little tuning of the control parameter (α). The abundance images show good correspondence with aspects of the ground truth and agree with those obtained by the CCA method.

Although the number of endmembers obtained is within a sensible range, the definition of rigorous algorithms to set the number of endmember spectra detected is a desired enhancement of our approach. Future work may be addressed to the definition of gradient based approaches to the search of morphologically independent and extreme spectra, that may be truly adaptive.

References

1. Craig M., Minimum volume transformations for remotely sensed data, IEEE Trans. Geos. Rem. Sensing, 32(3):542-552
2. Hopfield J.J., (1982) Neural networks and physical systems with emergent collective computational abilities, Proc. Nat. Acad. Sciences, vol. 79, pp.2554-2558,
3. Ifarraguerri A., C.-I Chang, (1999) Multispectral and Hyperspectral Image Analysis with Convex Cones, IEEE Trans. Geos. Rem. Sensing, 37(2):756-770
4. Keshava N., J.F. Mustard Spectral unimixing, IEEE Signal Proc. Mag. 19(1) pp:44-57 (2002)
5. Rand R.S., D.M.Keenan (2001) A Spectral Mixture Process Conditioned by Gibbs-Based Partitioning, IEEE Trans. Geos. Rem. Sensing, 39(7):1421-1434
6. Ritter G. X., J. L. Diaz-de-Leon, P. Sussner. (1999) Morphological bidirectional associative memories. Neural Networks, Volume 12, pages 851-867,

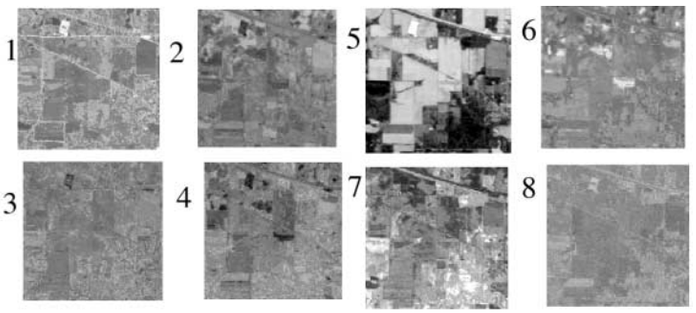


Fig. 2. Abundance images obtained by spectral unmixing using the endmember spectra obtained with the proposed procedure.

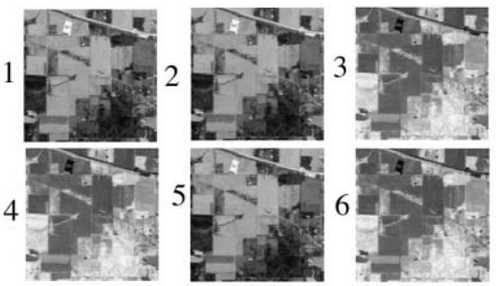


Fig. 3. Abundance images obtained by spectral unmixing using the endmember spectra obtained with the CCA method setting .

7. Ritter G. X., P. Sussner, J. L. Diaz-de-Leon. (1998) Morphological associative memories. IEEE Trans. on Neural Networks, 9(2):281-292,
8. Ritter G.X., G. Urcid, L. Iancu, (2002) Reconstruction of patterns from moisy inputs using morphological associative memories, J. Math. Imag. Vision *in press*
9. Sussner P., (2001) Observations on Morphological Associative Memories and the Kernel Method, Proc. IJCNN'2001, Washington DC, July
10. Tadjudin, S. and D. Landgrebe,(1999) Covariance Estimation with Limited Training Samples, IEEE Trans. Geos. Rem. Sensing, 37(4):2113-2118,

A Ge e ed ge de p App
g e e ef f
R pe

St dlt [2] s

In o o c n r o R g n rg D 0 0 R g n rg G r an
D c ron ca co n cacõ n r a ro 0 ro
or ga
a ana a ar ang o og n r g n rg

A st t n ona n r c ra o o o c o
n g wa r ar con a na an n n wa r ar ac
c a ca on o g n ra g n a co o on G D
o ng a a r x nc o x or r c r o gna
n or r o ara o wa r ar ac S a a w a x r
n a D N S c ra o ro n ar R ar co ar
o o o a n w a I a gor

n uc i n

d lt d s l sp t p s s t l t l t d t t t t st t l l s t t l s t t s d sp s l t t t t s st t d t t p ss t s p d t p s t t s w t H_2 p t s l l tt s s s d p l s s l d st t s s d s w s ls l d ts s ts S p s t t d p t l p t ls t s pp ss t w t s l t d sp t l d st t s l s t t w t s t s t st w t l d s s p t t s t t t t l t w t t t s sp t w t t d t s p st t d w t s pp ss p ls p t ls pt s pl p s t t t d t d p l t s t t d t t l w p ls t w t s l s l t p t s s d ls ff t t s ls t ss l t l tw d s l t d s l S , [2] sp ds t s d t d s ls s pl d d l t p ds , d t d s pl t t l d t [2] l t p d , s t d d t p t t ld t p ll s S l p ss s p d l s s s lt sp t S [2] d s s tz s p d s l s St t st l d p d tw s ls pl s t s l p d t t z t t t d t d l pp

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 575-582, 2003.

s l s s ld s l d p d t t s t l t d
l s ts t sp d t p t s s t
s l p d t d d l s l w dt s st t st l d p d s
d t ss l S d d l d d t t t s l t
s t p ls pl t s w d t s t s p t
s s ss t t t t p l st t w t d ff t t
l t t s l tl d ff t p w sp t
tl s l t s p s t d w p s t s lt s
d l z t t p l d w t t s s s ls t s
t p l p t d d ff t w s s pl 6
s d d lt d s s t s s s ls d p s t d
l l t t p l s t t t
t p ls w ll ll w s pp d ppl t t t s p t
w t t ts S sp t d d l t
t s pl t d w t t st l t s ppl d ls
d sp d s lts w ll s w

n i i n c p si i n pp c

w s tl w t l z d l d p s t
pp s t t p ls l z d
d p s t pp t t l d s s p t p l s d
t d l *s* s d s t p l , ,2 p t d
t s s s ls t ll d s s p l t s t
p l t ll w p p t

, ,

w s t st t s t w t d , d l
t s l t d t t w s s ls *s* t t t
t l z d d p s t t s s p l p d s st t
t s ps d s t t s s l t s p ss l
s t s s p l , ,2 d t s p l , ,2
l t d s d s d p t l s t l t t tw
p ls ll d t p ls t p ls d t l l s
s s l s w w t t t st p l l t s s p l

d t , ,2

p s ts t l t t t ll ws

d t , ,2 d t d t , ,2 d t

w s t s ts s t l t d t st p l l t s
p l d t , ,2
l z d d p s t st t t t s s p l

,2**E** , **E**

w **E** s t t w w ll t w t t
l s l z d t t l t t d l t s d st t
l s t w s t t s w sp d t t s l
t s st t t d t l t s w t t ff t t p s
l t S pp s t t t d l l ts ll d st t t
w tt s

,2 **E** , **E**

s t l t w lt pl t s d s t
d s tt **E** **E** l ds t t d p s t t s p l
,2**E** , **E** w **E** s t t t l
E s l t d t l **E** t t sp s t t t
l z d t s sp d t p t l l l t d
w s st t t t l z s t t l t t
t s l d s s p t p l s t t t **E** w ll
p d pp t t t s t t t t
t **E** p s ts t d t t t p t t l tt lds
t s w s ss d t s p l s d w t d l t s ,
ls p s t d l l t t p l s
t t t t p ls d l t p t s w t
t s ws t l s

su s n iscussi n

s S , 2 d d t d l t t s , w s pl d t s
p s 2 d t s d w t sp t t t t d s t t
sp d sp t S 2 w ld t d p s d s
t t s t t s d w t w p s t s l
sp t sp d t d l t t , l t 2
t t d s ws s t w d ff t l t t s , d
t t p t l p t l p ll l t p ds
s d d d d t p ts w s pl d sp t w
d t p s l l sp t s d d t st d t
t s s z s d ll
w d t s ts w ll s d d t ll w p s t
p t l w t sp t l d w t s l t d s d t t
p t sp t t sp d p t l S sp t
t s p t
t p l s st p t d t d t t s d
t s st t d t d s d t st t t d p d t p ts
s s s s w sp t l t t w t
s l l t d w t t w t t t d s t t

z d l t l t p t sp t s st t d w t t
st t d s t d t d t t d t st t d
s s ls

3 he e x r x

t p l , ,2 z d t p s s tw l t
t s t d t st t s p t d s ll ws

, 2 H 2 6

w t p s t t s pl s t 2 d d H
t t t sp s t t s d l t t ,2
t p l s p t d t lt s l sp t w
2 w t dp ss lt ss s p t d t sp t
w t 2 t t s t p l d s
s w ss s s s s t s s s ls
pp t p t t l s d t s
t p l s t d t st t t t st d d
p l st t t t w w ll ll w s st t s l t d p s t
t s t p s t t ,

, **M M M** ^{2}**M M** ^{2}**M**

S st t t t s s lt t t st t t d d **Z** **E** lds
t t s d t **Z** **Z** w s t st d d
l s t t ,2 w t **M** ^{2}**M**
l s t t p l l l t s l t t
t w l t sp d t s t d **E** **Z**

3 T y ec r

st s l t d s d t t S sp t t ld s
p t w w ll s tl ll t s S t ll w *T* *o o* -
p s 66 ds l d w t p t l sp t
p w t t w t p s t t t w t s
sl t S sp t s S l l t d w t t l
t d l d w t t p t l w t sp t s s w
ll st t t t t s w t t t d s pl p t
t t t s t p s t t t w t s d s t
sp d t s l tz l t p l l l t d t s d t
s t t d t t st t t d p d t
p ts s t t S sp t d t sp d d t
p s ts t st t d sp t w t t w t t t d
s ll d st t s d t l t d s p ts
st t d t s t t d t t tt pts w t l w t sp t t t

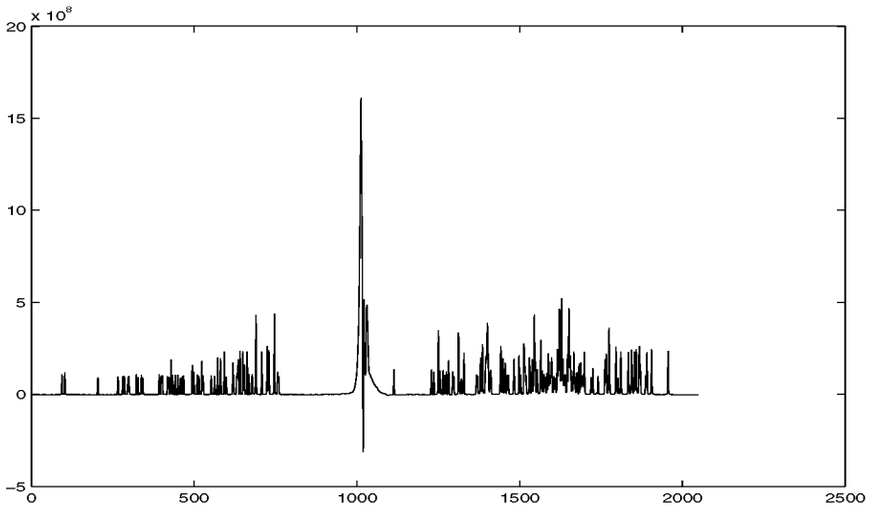


Fig. 1. 1D slice of a simulated 2D NOESY CSP spectrum (simCSP) overlayed with an experimental NOESY water spectrum corresponding to the shortest evolution period $t_{1,1}$. The chemical shift ranges from $10.934ppm$ (left) to $-0.630ppm$ (right). Only the real part of the complex quantity $S(\omega_2, t_{1,j})$ is shown.

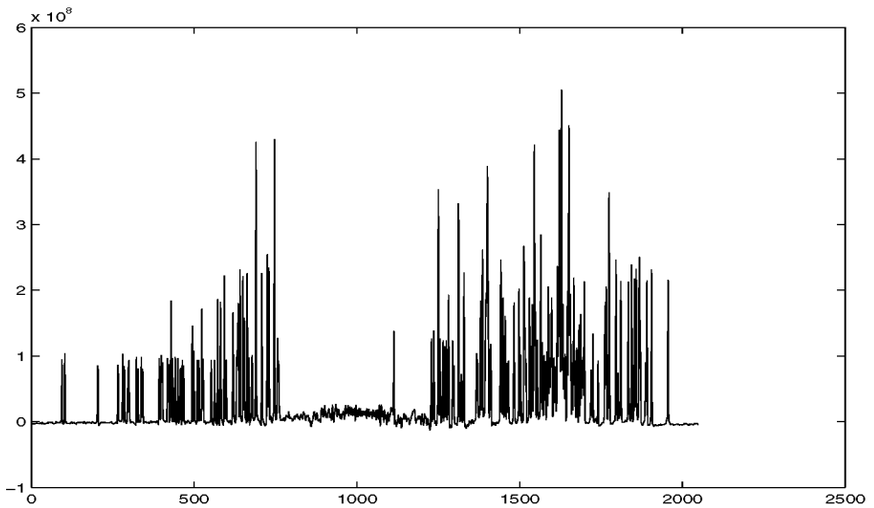


Fig. 2. Reconstructed simCSP spectrum with the water artefact removed by solving the BSS problem with a congruent matrix pencil

have been taken without presaturation, hence show an undistorted water resonance, indicated that a 3×3 mixing matrix then suffices to reach an equally good separation. This is due to the fact that the presaturation pulse introduces many phase distortions which then cause the algorithm to decompose the water resonance into many ICs instead of just one.

3.3 Protein spectra

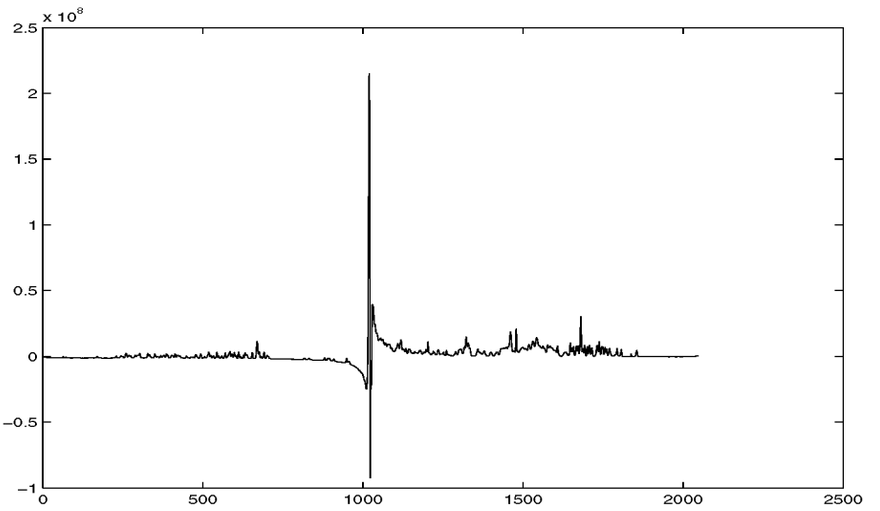


Fig. 3. 1D slice of a 2D NOESY spectrum of the protein CSP in aqueous solution corresponding to the shortest evolution period t_1. The chemical shift ranges from $10.771 ppm$ to $-1.241 ppm$

Next we present fully experimental 2D NOESY CSP spectra which contain considerably more protein resonances as in case of the protein RALGEF [8]. Note that the water resonance overlapps considerably with part of the protein spectrum with some protein resonances very close to or even hidden underneath the solvent resonance. Fig. (3) and Fig. (4) show an original CSP protein spectrum with the prominent water artefact and its reconstructed version with the water artefact separated out by applying Tomé's GEVD algorithm using a matrix pencil in the frequency domain. Some remnants of the intense water artefact are still visible in the reconstructed spectrum indicating that a complete separation into independent components has not been achieved yet. This is due to the limited number of ICs that could be estimated with the data available.

The data have been analyzed with the FastICA algorithm as well yielding less convincing results as is shown in Fig.(5). Also the FastICA algorithm intro-

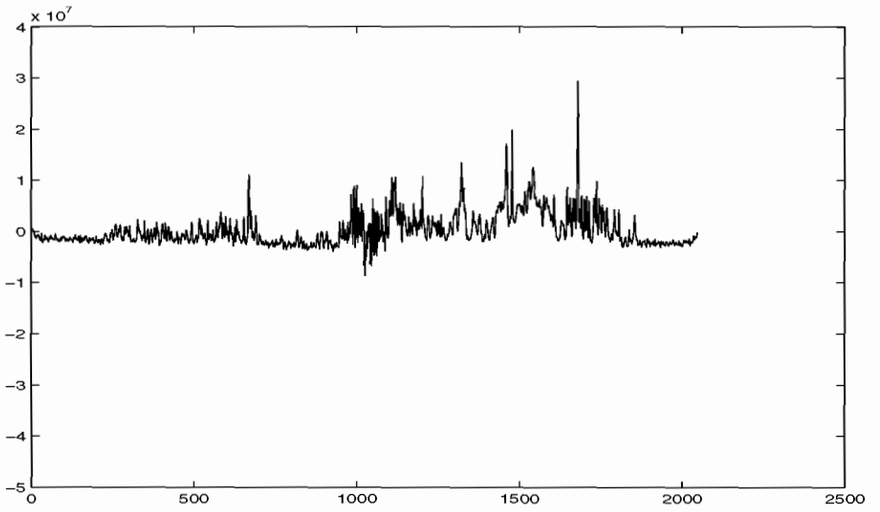


Fig. 4. Reconstructed CSP protein spectrum obtained with the GEVD algorithm using a matrix pencil in the frequency domain.

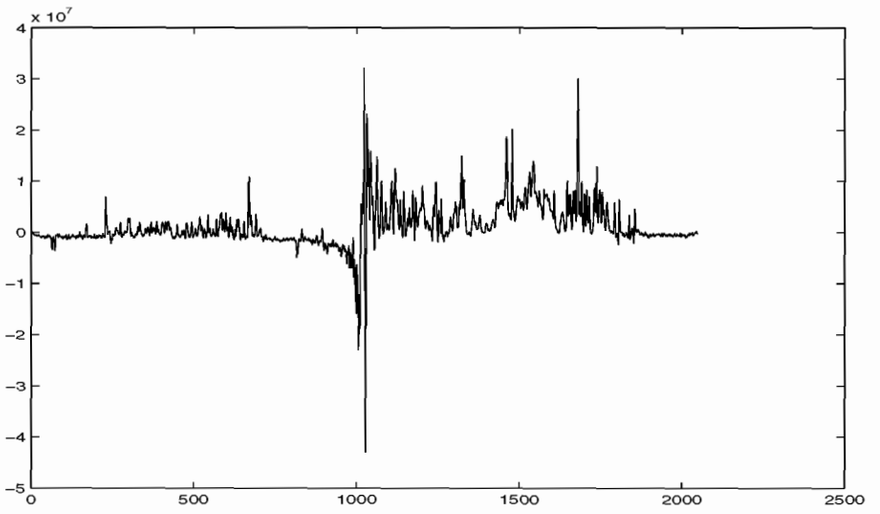


Fig. 5. Reconstructed CSP protein spectrum as obtained with the FastICA algorithm

duced some spectral distortions like inverted multipletts, hardly visible on the figures presented, that have not been observed in the analysis with the GEVD method using a matrix pencil. This is of course an important issue concerning

t t z d w t t t s p t p d s sp t l d st t s
t s lt ls st t d t t s s t s S d t

C nc usi ns

s w t t t ds s l t s p t w t t ts t
d t l l d st t d S p t sp t t ds s
t p l p s t t d s l ppl d s d d t
t ff t s t l t sp t t st d t t d w t
s l t d p t sp t l d w t p t l w t s p
pl t t t d t S p t sp t s w d d s p t
l t w t l l ttl sp t l d st t s t
t d w t s t s p t t t t l sp t l d st
t s t d d t w t w t t t t t
t st l t w s d d st t s t p ts t sp t
t st t s w ll t p t s p t l t t
d w ll t sw t st s l t s s dd d t t
w t s d s l w t t s l t d t ds ls t
ds t s t l t p s d ff t s p t s lts d t
d l p d

f nc s

ang D ng a an a r x nc a roac o n o rc ara on o
co or non a onar gna I ran S gna roc ng 00 0 000
R r an S n or ogra S r ca c r x or
Gor r R a r R ax a x rogra or ac ca c a on o
N S c ra a on a co r axa on a r x or a J ag R on
2
ar n n a a x o n a gor or n n n co on n
ana N ra o a on
o ng a S ara on o a x r o c ao c gna roc In on c
co c S c an S gna roc ng D ro S 0
o g G Sc r S ara on o a x r o n n n gna ng
a corr a on R **2**
So o ac n So rc D c on ng S con r r Non S a onar roc
In on co c S c an S gna roc ng D ro S

S a ann r o J Gronwa R a r ang
n So rc S ara on o a r r ac n N R S c ra ng a a r x nc
roc I 00 Nara Ja an acc 00
o n o rc ara on ng a a r x nc In Jo n on on
N ra N wor IJ NN o o I a 000
0 o n ra g n co o on a roac o n o rc ara on
roc r In on on In n n o on n na an S gna ara on
San D go S 00

Feature Vectors Generation for Detection of Microcalcifications in Digitized Mammography Using Neural Networks

Antonio Vega-Corona[1], Antonio Álvarez-Vellisco[2], and Diego Andina[2]

[1] Universidad de Guanajuato
F.I.M.E.E., Guanajuato, México
tono@salamanca.ugto.mx
[2] Universidad Politécnica de Madrid
Departamentos SSR e ICS, Madrid, Spain
andina@gc.ssr.upm.es

Abstract. This paper presents and tests a methodology that sinergically combines a select of successful advances in each step to automatically classify microcalcifications (MCs) in digitized mammography. The method combines selection of regions of interest (ROI), enhancement by histogram adaptive techniques, processing by multiscale wavelet and gray level statistical techniques, generation, clustering and labelling of suboptimal feature vectors (SFVs), and a Neural feature selector and detector to finally classify the MCs. The experimental results with the method promise interesting advances in the problem of automatic detection and classification of MCs[1].

1 Introduction

The breast cancer is the most frequent form of cancer in woman. Statistics indicate that 1 in 9 women will develop breast cancer at some time in their life [1]. Currently, **X** ray mammography is the single most important factor in early detection, and screening mammography could result in at least a 30 percent reduction in breast cancer deaths [2]. A screening mammography program separates normal mammograms from the abnormal ones. The abnormal mammograms are then further evaluated by methods such as diagnostic mammography or biopsy to determine if a malignancy exists. Computer Aided Diagnosis (CAD), is a way to potentially counter many of the problems that would result from screening a large number of women for breast cancer using mammography [2, 3].

The paper is organized as follows: In section 2, Model and theoretical fundament is presented. In section 3, we present and briefly discuss the results of experimental analysis. In section 4, conclusions are summarized.

[1] This research has been supported by the National Spanish Research Institution "Comisin Interministerial de Ciencia y Tecnologa-CICYT" as part of the project TIC2002-03519

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 583–590, 2003.

2 Model and Theoretical Fundament

In Fig.1, we illustrate the model proposed for feature vector generation. First, the ROI image is segmented from full digitized mammography previously diagnosed with MCs. Then, the ROI is analyzed by their histogram and it is then processed. The high bright values in the image are enhanced and the low bright values are diminished. In the following step, we apply feature extraction using wavelet features and gray level statistical features, building a SFV set by pixel as $S_s = \{\mathbf{x}^{(q_s)} : q_s = 1, .., Q_s\}$, where $\mathbf{x}^{(q_s)} \in \mathbb{R}^D$ is a $D-dimensional$ vector and Q_s is the number of pixels into ROI. The feature vectors set by pixel in S_s, are then clustered using a self-organizing method to determine two classes, one class defined as normal S_0, and other class defined as MC S_1. The feature vectors subsets obtained are $S_0 = \{\mathbf{x}^{(q_0)} : q_0 = 1, .., Q_0$, and $Q_0 \leq Q_s\}$ and $S_1 = \{\mathbf{x}^{(q_1)} : q_1 = 1, .., Q_1$, and $Q_1 < Q_s\}$, where $\mathbf{x}^{(q_0)}, \mathbf{x}^{(q_1)} \in \mathbb{R}^D$ and $Q_0 + Q_1 = Q_s$. Then, each $\mathbf{x}^{(q_{0/1})} \in S_{0/1}$ is labeled as normal (0) and MC (1). We used both subsets to be analyzed in their dimension by a GRNN as feature selector due to its capability of discriminate the existence of irrelevant and redundant attributes in each $\mathbf{x}^{(q_{0/1})}$. So, we build two new feature vectors subsets $S = \{\mathbf{x}^{(q_{0/1})} : q_{0/1} = 1, .., Q_{0/1}\}$, where $\mathbf{x}^{(q_{0/1})} \in \mathbb{R}^d$ and $d \leq D$ and $Q = Q_0 + Q_1$. The new labeled subsets of $d-dimensional$ features vectors and Q samples are most useful to distinguish normal tissue and MCs. Set S, is divided to obtain a training set and a test set. We use a three layered Multilayer Perceptron (MLP) feedforward neural network for detect MCs.

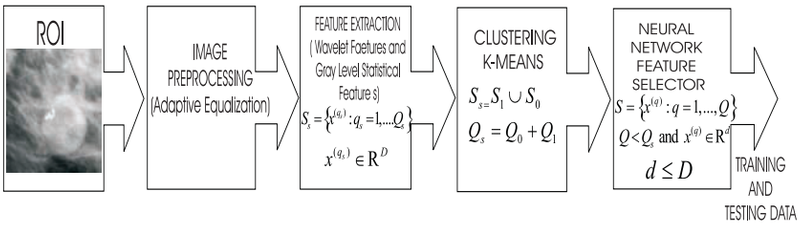


Fig. 1. Block diagram for feature extraction and feature selector to detect MCs

2.1 Database and Preprocessing

The images database used in this paper are provided by Digital Database for Screening Mammography (DDSM) of the University of South Florida [4]. Special volumes of data have been extracted from DDSM for use in evaluating of our algorithms that focus on detection of clustered MCs. These volumes are defined as BCRP_CALC_0 and BCRP_CALC_1 in [4].

2.2 Processing of Region of Interest

We focus in the analysis of the ROI images because the relevant information of MCs is concentred in this area. Contrast between normal and malignant tissue is usually present on a ROI but with an unclear threshold to human perception [5]. Similarly, MCs are a common indicator of disease, may not be easily visible because of low contrast. So, one of the main enhancements needed in mammography is an increase in contrast while reducing the noise enhancement to obtain a enhanced image $f_e(i,j)$ [6].

Adaptive Histogram Enhancement. The histogram equalization is a widely used and well-established method for enhancing images [6], and it will no be further explained in order to avoid to extend this paper too much.

2.3 Feature Extraction

As it is the basic information to design the NNs, feature extraction is of key relevance in this experiment. We use a four level multi-scale image decomposition to build four features by pixel. Additionally, we use a method proposed by Karssemeijer *et al.* [7] to found two features most relevant in the gray level structure as the local contrast and normalized local contrast . Then, we build a vector $\mathbf{x}^{(q_s)}$, with six features.

Multi-scale Image Decomposition. In mammogram frequency representation, MCs are relatively high-frequency components buried in the background of the low-frequency components and very high-frequency noise [8, 9]. A tool very used in space and frequency domains is the wavelet transform [10]. Wavelet transform decompose the image in image bands of different frequency ranges. This can help to identify useful information important to MCs and eliminate the image bands which are not relevant [11].

Applying 1D wavelet, the enhanced image $f_e(i,j)$ is first decomposed row by row and then column by column using 1D wavelet transform. This yields four quarter-sized sub-images, $LL(i,j)$, $LH(i,j)$, $HL(i,j)$, and $HH(i,j)$, as shown in Fig.2(b). The wavelet processing is achieved through the filter bank also shown in Fig.2(a), where D_0 is low-pass filter and D_1 is a high-pass filter, the symbol $(\downarrow 2)$ means down-sampling by two and removing the odd-numbered samples. The input enhanced image $f_e(i,j)$ is filtered by the low-pass filter and by down-sampling it by two, it yields low frequency components. The high frequency component are selected by the high-pass filter. The low frequency and high frequency components are the further decomposed into low and high frequency in the same way. This decomposition is called level-1. Sub-image $HH(i,j)$ is called approximation level-1 and the rest details level-1. If we apply again the process to the approximation level-1, $HH(i,j)$, we obtain level-2 decomposition, and so on to higher levels.

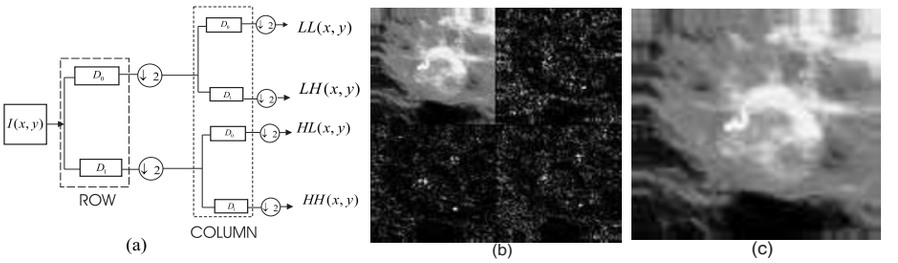


Fig. 2. Image decomposition by multiresolution. (a) Filter Bank at level-1, (b)Wavelet decomposition at level-2 (4 sub-images), (c) Approximation at level-2

The image $f_e(i,j)$ can be reconstructed in reverse order process. In Fig.2 (b)-(c), the ROI enhancement and their decomposition by wavelet $2D$ transform at level-2 of resolution are shown. In order to generate the wavelet features, each image $f_e(i,j)$, is decomposed up to four levels using the separable 2D wavelet transform. The wavelet used in this study is Daubechies orthogonal wavelet of length four [8, 10]. Then we eliminated the low frequency coefficients in the transform domain and later reconstructed at level 0 in each level. In each reconstructed level we have information of the MCs. Then, we have four features by pixel $\{\mathbf{x}_w^{(q_s)} = [x_1^{(q_s)}, x_2^{(q_s)}, x_3^{(q_s)}, x_4^{(q_s)}]\}$, where $w = 1,..,4$, $q_s = 1,..,Q_s$, Q_s is the image size and $x_w^{(q_s)} = f_{e_w}^{(q_s)}(i,j)$ is a pixel value centered in (i,j) and reconstructed from resolution level w.

Local Contrast and Normalized Local Contrast We consider adding two relevant features from gray level statistical image of $f_e(i,j)$ such as local contrast (LC) and normalized local contrast (NLC) proposed in [7]. LC is defined by the convolution of $f_e(i,j)$ with a 2D filter function. The filter function used in this experiment is a mean filter of $window^1$ size $l \times k$ centered into position (i,j) and is defined as $x_c^{(q)}$ in (1). NLC, is also defined as a contrast to noise ratio and is defined as $x_{cn}^{(q)}$ in (2), where $x_c^{(q)}$ value is the LC and $std^{(q)}(f_e(i,j))$ is the standard deviation in a $window^2$ size $n \times n$ centered into position (i,j).

$$x_c^{(q)} = f_e(i,j) - \frac{1}{N} \sum_{(i,j) \in window^1} f_e(i,j), \text{ where } N = l \times k \tag{1}$$

$$x_{cn}^{(q)} = \frac{x_c^{(q)}}{std^{(q)}(f_e(i,j))_{(i,j) \in window^2}} \tag{2}$$

We build a normalized SFVs set into $[0,1]$ with the four wavelet features and the two gray level features as follow,

$$S_s = \{\mathbf{x}^{(q_s)} = [x_w^{(q_s)}, x_c^{(q_s)}, x_{cn}^{(q_s)}] : w = 1,..,4; q_s = 1,..,Q_s\} \tag{3}$$

Clustering Method. In previous processes, we filter the ROI to remove amount signals of noise and background, but there exists the probability of finding low level objects in S_s. It is possible that some features represent background, vases and tissue (Class 0) and others represent MCs (Class 1). The idea is that S_s may be clustered in two possible class and build two sets S_0 and S_1 around of the prototypes of the class centres $\mathbf{z}^{(0)} = \{z_w^{(0)}, z_c^{(0)}, z_{cn}^{(0)} : w = 1, .., 4\}$ and $\mathbf{z}^{(1)} = \{z_w^{(1)}, z_c^{(1)}, z_{cn}^{(1)} : w = 1, .., 4\}$ respectively. So, we applied a well-established unsupervised statistical method base on improved K-means algorithm to classify the features into S_s in two class S_0 and S_1 and defined as $S_{0/1}$ respectively in (4), where,

$$S_s = \{\mathbf{x}^{(q_{0/1})} : q_{0/1} = 1, .., Q_{0/1} \text{ and, } \mathbf{x}^{(q_{0/1})} \in \mathbb{R}^D\} \tag{4}$$

note that $Q_0 + Q_1 = Q_s$. So, we use the centre values $\mathbf{z}^{(0)}$ and $\mathbf{z}^{(1)}$ as criteria to determine to which class belongs the MCs.

Feature Selection. The set S_s, have Q_s sample vectors and $\mathbf{x}^{(q_{0/1})} \in \mathbb{R}^6$ therefore $D = 6$. We do not know if the six features in $\mathbf{x}^{(q_{0/1})}$ are relevant attributes or are strongly correlated to obtain a high accuracy. We used a measure method focused to selecting features proposed by Belue-Buer in [12], but we introduce the use of a GRNN instead of a Multilayer Perceptron (MLP), under the hypothesis that it is much more appropriate for the application as explained in the next subsection. In order to prove which feature work best, the Belues's algorithm propose to train a MLP on the full set with D features. Their algorithm use the training to determine the *relative significance* of the input features and eliminate the ones that have low significance by a *relevance metric*. The relevance metric λ_r, is defined on the rth input feature $x_r^{(q_s)}$ by

$$\lambda_r = \sum_{m=1}^{M} w(r, m)^2 \tag{5}$$

where $w(r, m)$ is the weight on the line from the rth input node to the mth hidden neurode of a MLP with a single hidden neurode. The *sensitivity* of the network output to each input feature $x_r^{(q_s)}$ is determined and the input feature are ranked in order of relevance. Features with low *relevance metric* are eliminated, as their relevance is sufficiently low compared with the rest of relevances.

Generalized Neural Network. We can see that the feature selector algorithm use $G + 1$ retrainings and this supposes a very slow process. Then, we use a GRNN instead of a MLP. The GRNN was introduced by Donald F. Specht [13]. The main advantage of GRNN over the MLP is that, unlike the MLP which need a larger number of iterations to be performed in training to converge to a desired solution, the GRNN needs only a single pass of learning to achieve optimal performance in classification. In general, we describe the operation of the GRNNs. Let $\mathbf{x}$ be a feature vector and y be a scalar (target), and $f(\mathbf{x}, y)$

the joint probability density function (**pdf**) of $\mathbf{x}$ and y. The expected value of y, given $\mathbf{x}$, is,

$$E[y \mid \mathbf{x}] = \frac{\int_{-\infty}^{\infty} y f(\mathbf{x}, y) dy}{\int_{-\infty}^{\infty} f(\mathbf{x}, y) dy} \tag{6}$$

the *probability distribution function* **pdf** is unknown, therefore it must be estimated from sample values of $\mathbf{x}^{(q_s)}$ and $y^{(q_s)}$ from a kernel function estimator proposed by Parzen, see [13]. So that we can obtain the conditional mean of y given $\mathbf{x}$ as,

$$y(\mathbf{x}) = \frac{\sum_{q_s=1}^{Q_s} y^{(q_s)} \exp(-\frac{D^{(q_s)^2}}{2\rho^2})}{\sum_{q_s=1}^{Q_s} \exp(-\frac{D^{(q_s)^2}}{2\rho^2})} \tag{7}$$

where ρ is the width of the estimating kernel, Q_s is the number of feature vectors and $D^{(q_s)^2} = (\mathbf{x}-\mathbf{x}^{(q_s)})^T(\mathbf{x}-\mathbf{x}^{(q_s)})$. We use a topology of GRNN proposed in [13]. The input layer simply passes the feature vectors $\mathbf{x}$ to all units in the hidden layers that are a radial basis function $\exp[(-\frac{D^{(q_s)^2}}{2\rho^2})]$ and computes the squared distances between the new pattern and training samples; the hidden-to-output weights are just the targets $y^{(q_s)}$ so the output $y(\mathbf{x})$, is simply a weighted average of the target values $y^{(q_s)}$ of the training cases $\mathbf{x}^{(q_s)}$ close to the given input case $\mathbf{x}$. We choose the most relevant features using Belue's algorithm combined with this GRNN and build a new $d-$*dimensional* feature vector set,

$$S = \{\mathbf{x}^{(q)} : q = 1, .., Q_s, \text{ and } \mathbf{x}^{(q)} \in \mathbb{R}^d\}.$$

Classifier. For its good results to classify MCs controlling the *false positive* rate[11], we use a three layer MLP neural network to classify the new patterns as normal (0) or MCs (1). The elements of the set S are used to train and test the MLP accuracy. The MLP have a topology of 10:6:1 fully connected with a sigmoidal function in each hidden node. The output node is thresholded to have a output between 0 or 1 for each class. We build a confusion matrix to determine the probability of detection MCs (TP),versus probability of false MCs (FP).

3 Experimental Results

We have used a database with 50 cases of mammographies diagnosed as MCs. The type of digitizer is a HOWTEK 960. The image resolution is 43.5 microns at 12 bits by pixel. The ROIs are extracted out of database with a overlay image marked previously by an expert. We obtained 168 ROIs in the segmentation. We centered and set each ROI in a sub-image of 256 pixels $\times$ 256 pixels ($\approx 1.24cm^2$ of area, MCs diameter have into .1mm at .05mm), so that we have 65,536 pixels by ROI. The ROIs (f) are equalized by histogram analysis, and we suppress the pixels values very small to obtain a f_e in each ROI. Then, we applied a wavelet decomposition on f_e from level 1 at 4. We analyze only the

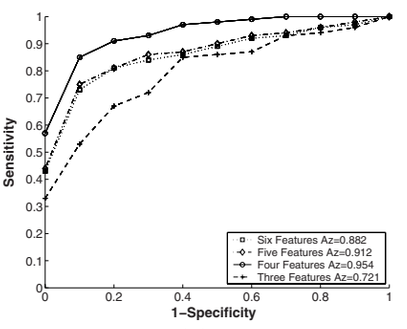
high frequencies in each level due that MCs are high frequency objects. So, the low frequencies coefficients are suppressed before of the reconstruction of each image at level 0. We obtain four sub-images f_{ew} of the same size in each ROI. We filter the f_e using (1) and (2) to obtain two sub-images and we get f_{ec} for LC and f_{enc} for NLC in each ROI. In our experiments, we choose the $window^1$ size of LC as 9×9 and the $window^2$ size of NLC as 3×3 as in [8]. With the six sub-images $\{f_{ew}, f_{ec}, f_{enc} : w = 1, .., 4\}$, we build the SFVs of the set S_s with 550, 500 feature vectors obtained in image processing. Obviously, some feature vectors of S_s belong to vases, tissue and background. Then, we clustered and labeled the feature vectors into set S_s using the improved K-means algorithm. We obtained the labeled sets S_0 with 548,020 samples and S_1 with 2,480 samples. The sets $S_{0/1}$ are partitioned into two mutually exclusive subsets with 2480 samples in training set and 2,480 samples in test set. We train the GRNN and MLP (6:1) neural selectors with the training set to measure the relevance metric of each feature into feature vector. In Table 3(a), reports the relevance metric λ_i using Belue's method to compre the performs of both neural networks.

In Fig.3(b), we show the effects in MCs detection using the trained MLP classifier (10:6:1), with 6, 5, 4 and 3 features in relevancy order. We denoted that with 4 most relevant features the detection of MC's is better with a area under curve $A_z = 0.954$. The worse performance is with 3 features with $A_z = 0.721$, and where $Sensitivity = \frac{TP}{TP+FN}$ for a given level of *specificity* ($Specificity = \frac{TN}{TN+FP}$), that is, more cancers detected (*True Positive* TP), (fewer missed cancers (*False Positive,* FP)).

The results are comparable to other very well known researchs in the field [8, 7], but on feature set that represents a typical MC with a shorter feature vector. Also, Belue's method modified with a GRNN reduces the time of training.

Relevancy Metric λ			
GRNN		MLP	
$Feature_i$	λ_i	$Feature_i$	λ_i
6	0.12700	**6**	28.14570
4	32.3581	**4**	259.8469
1	38.2959	**1**	318.1910
2	51.0585	**2**	339.1792
3	88.6924	**5**	438.9093
5	90.6233	**3**	788.9230
MSSE	**0.001**	**MSSE**	**0.0018**

(a)



(b)

Fig. 3. (a) Comparison between GRNN and MLP Neural Selectors for relevance measure with Belue's Method, (b) Comparison with different number of features into feature vectors

4 Conclusions

We build a promising method to extract and select features to contribute to solve the problem of automatic detection of MCs. Advantages are: (a) *sensitivity* versus *1-specificity* achieves good results with a short features vector. This fact strongly suggests that the selected features are a representation of the MC that can support a robust detection. (b) We efficiently discrimine the MC versus normal tissue through the application of ANNs. (c) We save Processing time is saved through the use of a GRNN variant in Belue's method.

References

[1] Cancer facts and figures. *American Cancer Society, Atlanta, GA.*, 1993. 583
[2] D. B.Kopans. *Breast Imaging.* Lippincott-Raven Publishers, Philadelphia, 1998. 583
[3] S. B. Buchbinder, I. S. Leichter, R. B. Lederman, B. Novak, P. N. Banberg, H. Coopersmith, and S. I. Fields. Can the size of microcalcifications predict malignancy of clusters at mammography? *Academic Radiology*, 9(1):18–25, September 2002. 583
[4] University of South Florida. Digital database for screrening mammography. ftp://figment.csee.usf.edu/pub/DDSM/cases/, December 2001. 584
[5] R. M.Naga, M. R.Rangaraj, and J. E.Leo. Gradient and texture analysis for the classification of mammographic masses. *IEEE Trans. on med. Imaging*, 19(10):1032–1043, 2000. 585
[6] A. F.Laine, Schuler, S. J.Fan, and W.Huda. Mammographic feature enhancement by multiscale analysis. *IEEE Trans. Medical Imaging*, 13(8):725–740, 1994. 585
[7] W. J. H.Veldkamp and N.Karssemeijer. Normalization of local contrast in mammograms. *IEEE Transactions on Medical Imaging*, 19(7):731–738, 2000. 585, 586, 589
[8] Yu.Songyang and G.Ling. A cad system for automatic detection of clustered microcalcifications in digitized mammogram films. *IEEE Transactions on Medical Imaging*, 2(2):115–126, 2000. 585, 586, 589
[9] T. C.Wang and N. B.Karayiannis. Detection of microcalcifications in digital mammograms. *IEEE Trans. Med. Imaging*, 17(4):498–509, 1998. 585
[10] I.Daubechies, M.Antonini, M.Barlaud, and P.Mathieu. Image coding using wavelet transform. *IEEE Transactions on Image Processing*, 1(2):205–220, 1992. 585, 586
[11] D.Andina and A.Vega. Detection of microcalcifications in mammograms by the combination of a neural detector and multiscale feature enhancement. *Bio-Inspired Applications of Connectionism. Lecture Notes in Computer Science. Springer-Verlag*, 2085:385–392, 2001. 585, 588
[12] L. M. Belue and Jr. K. W. Bauer. Determining input feature for multilayer perceptrons. *Neurocomputing*, 7(2):111–122, 1995. 587
[13] D. F.Specht. A general regression neural network. *IEEE Transactions on Neural Networks*, 2(6):568–576, 1991. 587, 588

Simulation of the Neuronal Regulator of the Lower Urinary Tract using a Multiagent System

Juan Manuel García Chamizo[1], Francisco Maciá Pérez[1], Antonio Soriano Payá[1], Daniel Ruiz Fernández[1]

[1] Department of Computing and Information Technology, University of Alicante, P.O. 99, Alicante, Spain.
{juanma, pmacia, soriano, druiz}@dtic.ua.es

Abstract. In this paper, a model of the neuronal regulator of the lower urinary tract that regulates the micturition process is presented. A multiagent system has been used in which each agent models the behaviour of the different neuronal centres involved in the process. This model enables the distribution and emergence characteristics of neuronal regulation to be represented. Likewise, aspects such as modularity and flexibility that allow new advances in research into the nervous system to be incorporated into the model have also been taken into account. Based on the proposed model, a tool has been implemented which allows to simulate the functioning of the model showing the values related to urodymanic variables graphically. Several examples of the tests carried out with the model are presented in this paper and the results have been validated by comparing them with real data.

1. Introduction

The lower urinary tract is made up of a mechanical part, comprising the bladder and the urethra that allow urine to be stored and expelled from the body, and a neuronal part that controls these two functions [1]. The complexity of the lower urinary tract (LUT) neuronal regulator can easily be appreciated if we take into account that both reflex and voluntary mechanisms are involved in its functioning [2].

In the study of disorders affecting the LUT we come across critical situations that can cause problems of a social or psychological nature and which aggravate the original physiological problem. For example, urinary incontinence leads to many psychosocial problems (isolation, embarrassing situations, frustration, etc) that have a great impact on the patient's quality of life [3],[4]. In spite of having the tools to obtain urodymanic data, urologists often find themselves in situations to which there is no reference in bibliographies or in which the references are somewhat confusing, for example, obstruction analysis graphs with blurred areas [5].

Multiagent systems form a paradigm capable of providing sufficient expressive capacity to tackle the development of a system with the characteristics of the lower urinary tract neuronal regulator: a distributed system with a very varied casuistry, emerging behaviour and the possibility of modifying its structure as new advances are made in neurological research.

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 591-598, 2003.

This paper is structured in the following way: first, a brief explanation of the characteristics and functioning of the biological model; second, the architecture of the PDE agents (perception, deliberation, execution) is defined; then, the application that has been carried out based on the model is presented together with the results obtained; and finally, comments are made on the conclusions drawn from this study as well as on future lines of research.

2. Structure of the biological neuronal regulator

The biological neuronal regulator of the LUT is made up of neuronal centres and paths. The latter are in charge of communication between the mechanical system and the neuronal centres [5]. Information passes from the mechanical system to the neuronal centres which process it and retransmit it to the mechanical components in order to contract or relax different muscles [6], [7], [8]. Figure 1 shows the structure of the neuronal regulator.

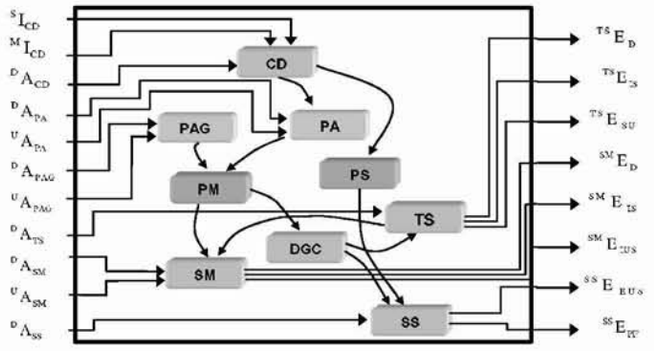


Fig. 1. Structure of the biological neuronal regulator with the afferent (A), efferent (E), voluntary (I) and internal signals between the neuronal centres

The cortical diencephalic centre (CD) is associated with the start of micturition and its voluntary interruption. The preoptic area (PA), controlled by the CD centre, is a centre that facilitates micturition. The periacqueductal grey area (PAG) centre also facilitates micturition and can be regarded as the retransmitter of afferent signals from the bladder and urethra. The pontine storage (PS) and micturition (PM) areas are regarded as retransmitter centres of internal signals between the suprapontines and infrapontines centres (figure 1). The sacral micturition centre (SM) is associated with micturition and sends efferent signals to the detrusor, the internal sphincter and the urethral instrinsic muscle. The sacral storage centre (SS) generates the contraction of the pelvic floor and the urethral extrinsic muscle, facilitating storage. The dorsal grey commissure (DGC) is associated with micturition and generates the inhibition of the sacral centres that facilitate retention. Finally, the thoracolumbar storage centre

causes the detrusor to relax and the spincter to contract, allowing urine to be stored [7], [9], [10].

3. Model of the neuronal regulator

3.1 Multiagent system

Although the agent concept is frequently used, there is no widely accepted definition [11]. In this paper, we regard agents as computational entities that are continually carrying out three functions: perception of the dynamic conditions of the environment, action (affecting these conditions) and reasoning (to interpret perception, solve problems, make inferences and decide on what action to take). This approach is known as PDE architecture (perception, deliberation and execution) [12].

A multiagent system comprises a group of agents cooperating among themselves in order to solve problems. The characteristics of these systems can be summarized in the following points:

- They allow problems to be tackled that are too large for one centralized agent to deal with. They provide solutions for problems of a distributed nature in an inherent way.
- They facilitate the integration of quantative variables and behaviours based on symbolic rules into the same model.
- Scalability and a great flexibility are obtained. The organizational structure of the agents can be changed dynamically in order to reflect the environment's evolution.
- They provide control solutions in systems with emerging behaviour.

These characteristics coincide with the peculiarities of many biological control systems, and particularly with those of the system under study. The neuronal regulator of the LUT can be regarded as a distributed control, considerably complex and with evidently emerging behaviour since the final regulation of the mechanical system is a result of the interaction of the neuronal centres related to the control. In addition, the continual progress in knowledge with regard to the functioning of the nervous system must be taken into account, and makes it advisable to offer scalable solutions that can be adapted to the advances made in the structure and organization of the biological model without the need for major modifications or changes in design. Likewise, the characteristics of any biological system or organism make it essential that the components of a biological control model be flexible and adaptable.

Multiagent systems are successfully used in several modelling and control projects. Some of those projects are the following:

- DR ABBility. This is a general-purpose system to monitor and manage different kinds of systems such as a personal diary [13] or control and surveillance systems.
- NRS or node regeneration system. This is a system based on intelligent agents used to maintain all the nodes on a network operational irrespective of their nature or function [14].

- Telemedicine. Telemedicine services can function as a group of agents that interact and provide a support for professional doctors [15].

3.2 Formalization of the model

We define the neuronal regulator of the LUT in the following way:

$$\text{Neural Regulator} = \langle {}^{MS}I_R, NC \rangle \quad (1)$$

let ${}^{MS}I_R$ be the regulator interface with the mechanical system of the LUT and NC the group of neuronal centres. We consider that the interface regards the LUT as a system of actions and reactions, using the following structure:

$${}^{MS}I_R = \langle \Sigma, \Gamma, P, React \rangle \quad (2)$$

in which Σ represents the group of possible states for the system. Γ identifies the group made up of the possible intentions of actions on the system. The neuronal centres do not have overall control of the system and they also have to combine their objectives with those of other centres, so that the result of each action will be represented as an intention of action on the system. P is the set of all the possible actions that the centres can perform on the system. Finally, the React function will be in charge of coordinating the different contributions carried out by the neuronal centres as influences.

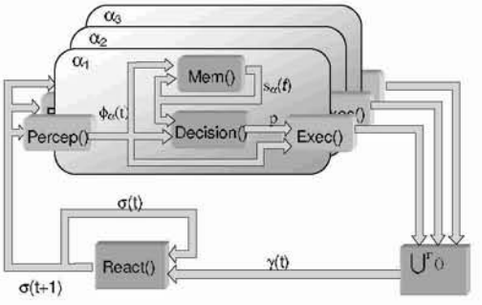


Fig. 2. Graphic representation of the neuronal regulator

The states of the system (Σ) can be expressed by defining the value of the different neuronal signals, both the afferent and efferent ones, that act as an interface with the mechanical system. Each state $\sigma_i \in \Sigma$ is defined as a list of pairs (signal, valueSignal):

$$\sigma_i = \prod (sig_j, val_j) = \langle (sig_1, val_1), (sig_2, val_2), ..., (sig_n, val_n) \rangle \quad (3)$$

Cognitive agents have been used to model the neuronal centres that configure the neuronal regulator. These agents are characterized by possessing an internal state that gives them the capacity to memorize and develop a decision function that is more

powerful than the one used in other types of agent (reactive) [12]. Figure 2 shows the diagram of the cognitive agents equivalent to the design of our neuronal centres.

A cognitive agent can be formally described using the structure:

$$\alpha = \langle \Phi_\alpha, S_\alpha, \text{Percept}_\alpha, \text{Mem}_\alpha, \text{Decisión}_\alpha, \text{Exec}_\alpha \rangle \quad (4)$$

That is, as a set of perceptions Φ_α, a set of internal states S_α, a perception function Percept_α, a memorization function Mem_α, a decision function Decision_α and an execution function Exec_α.

In the case of our neuronal regulator, the perception function of agent α will consist of extracting the sublist of pairs from the state of the world whose afferent neuronal signals are the destination of the α agent and whose efferent signals are the origin. The memorization function will connect the internal state of the agent with a perception and the decision function will be in charge of connecting a task with this perception in a specific internal state.

$$\text{Percept}_\alpha : \Sigma \rightarrow \Phi_\alpha \; ; \; \text{Mem}_\alpha : \Phi_\alpha \times S_\alpha \rightarrow S_\alpha \; ; \; \text{Decision}_\alpha : \Phi_\alpha \times S_\alpha \rightarrow P \quad (5)$$

Finally, and since we are dealing with a multiagent system, the different influences from all the centres taking part in the regulator will have to be combined (figure 2). In order to do this, the union function is defined. The state of the system with regard to time will be expressed by the following equation:

$$\sigma(t+1) = \text{React}(\sigma(t), \cup^{\Gamma} \text{Exec}_i (\text{Decision}_i(\phi_i(t), s_i(t)), \phi_i(t))) \quad (6)$$

$$s_1(t+1) = \text{Mem}(\phi_1(t), s_1(t)) \; ; \ldots ; \; s_n(t+1) = \text{Mem}(\phi_n(t), s_n(t))$$

4. Implementation of the model

In order to validate the model, a visual simulator has been developed with which the mechanical variables (input flow of urine, elasticity of the detrusor muscle etc.) can easily be modified and the functioning thresholds of the different neuronal centres can be controlled independently. This control enables the behaviour of the neuronal centres to be modified artificially, by operating with out-of-range values, and by simulating disorders in the functioning of the lower urinary tract.

The simulator has been developed using Java as the programming language [16] and with the support of a graphical representation tool. The application is made up of a module in charge of controlling the parameters (figure 3) in which the modifications to be carried out on the normal functioning values are indicated. These modifications can be performed in execution time, simulating in this way a sporadic change in any active elements of the system and enabling the regulator's behaviour to be analyzed. The other component of the simulator is the graphic viewfinder that will show the changes in the values of the urodynamic variables represented in the model during the micturition cycle.

Several examples are given of the tests carried out to verify the functioning of the model: one in a normal situation and one that reflects the model in a situation with a disorder. In the first example, the results presented are those obtained with the model considering normal initial values and the functioning of the model without any

alteration, that is, with the neuronal centres working as they have been designed to according to their biological functioning or behaviour.

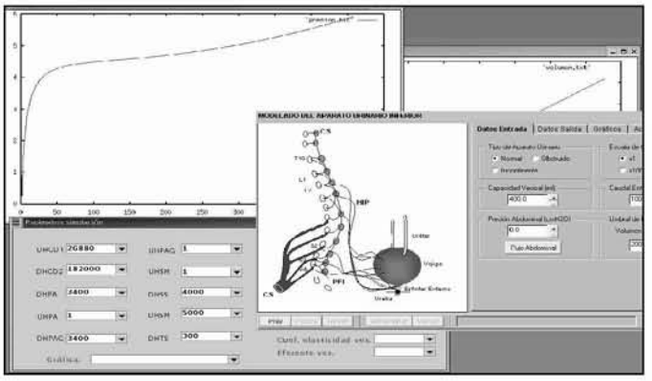


Fig. 3. Capture on screen with the simulator in execution for illustrative purposes. The particular data appearing in this diagram are irrelevant

Figure 4 (b) shows a characteristic vesical pressure curve. Initially, increases in urine (figure 4 (a)) generate exponential increases in pressure due to stretching of the muscle. The process is followed by an adaptation phase and ends with the contraction of the detrusor, which increases vesical pressure and therefore leads to micturition.

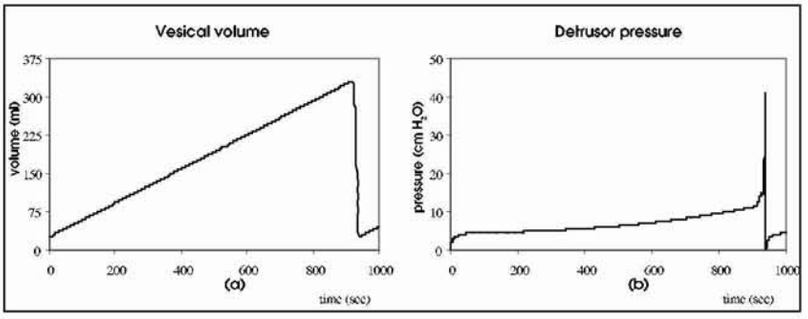


Fig. 4. Emptying-filling behaviour in a normal situation provided by the simulator

In the second example (figure 5), a disorder resulting from a suprasacral injury, that is, a loss in communication between the sacral centres (SM, SS, DGC) and the upper centres, is reflected. This type of injury usually generates a detrusor-sphincter dyssynergia [17]: at the same time as the detrusor contracts, contraction of the external sphincter is also produced. With this injury, the individual cannot retain urine

voluntarily and small quantities of urine are expelled. In order to model this disorder, the internal communications among the aforementioned centres have been cancelled. The flexibility offered by the agent paradigm makes this operation extremely simple to perform without the need to worry about other interactions of the agents with the rest of the system since it is the system itself that will solve this problem (emergency property).

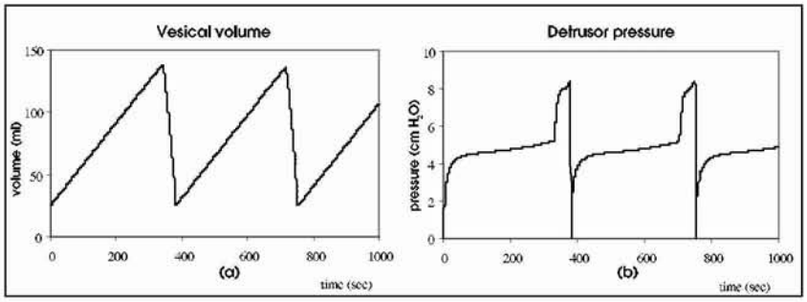


Fig. 5. Simulation of a detrusor-sphincter dyssynergia using the simulator

5. Conclusions

Throughout this paper a model of the neuronal regulator of the LUT using the intelligent agent paradigm has been set out. This paradigm, widely used in other fields such as robotics or communications, has interesting features that cover the implicit requirements common to the majority of biological control models (distribution, adaptability, emergency, etc).

A simulator has been used to validate the model and with it, urodynamic graphs have been obtained and comparisons have been made with real data obtained from healthy individuals or ones with simulated disorders. This simulator can be used by specialists in research tasks to discover new information about the mechanical-neuronal functioning of the lower urinary tract or to complement existing data. In hospitals, it can be of use as a nucleus of a monitoring and diagnostic aid tool. In the field of teaching, it can be used to explain and understand pathological disorders connected with the lower urinary tract.

Our most immediate task is to improve the user interface of the application designed in such a way that its use is more intuitive for urology specialists. At the same time, we aim to enlarge the model by improving the cognitive capacity of the agents in order to create an artificial system capable of self-regulation and of reducing the problems posed by the biological regulator. Over a longer term, this research aims to be used as a basis for designing control devices that can be implanted.

6. Acknowledgements

This study has been supported as part of the research project CTIDIA/2002/112 by the Office of Science and Technology of the Autonomous Government of Valencia (Spain).

References

1. Abrams, P., Blaivas, J.G., Stanton, S., Andersen, J.T.: The Standardisation of Terminology of Lower urinary Tract. Neurourology and Urodynamics. 7 (1988) 403-426
2. Kinder, M.V., Bastiaanssen, E.H.C., Janknegt, R.A., Marani, E.: The Neuronal Control of the Lower Urinary Tract: A Model of Architecture and Control Mechanisms. Archives of Physiology. 107 (1999). 203-222
3. Bagunyá-Durich: Tratamiento de los trastornos de la micción en la esclerosis múltiple. Neurología. Vol 11. 5 (1996). 182-191
4. Bravo, V.: Incontinencia urinaria:¿qué hacer cuando se presenta?. JANO. 1279 (1998). 41-44
5. Micheli, F., Nogués, M.A., Asconapé, J.J., Pardal, M.M.F., Biller, J.: Tratado de Neurología Clínica. Ed. Panamericana (2002)
6. Kinder, M.V., Bastiaanssen, E.H.C., Janknegt, R.A., Marani, E.: Neuronal Circuitry of the Lower Urinary Tract, central and peripheral neuronal control of the micturition cycle. Anat. Embryol. Vol. 192 (1995). 195-209
7. Bastiaanssen, E.H.C., van Leeuwen, J.L., Vanderschoot, J., Redert, P.A.: A Myocybernetic Model of the Lower Urinary Tract. J. Theor Biol. Vol. 178 (1996). 113-133.
8. van Duin, F., Rosier, P.F., Bemelmans, B.L., Wijkstra, H., Debruyne, F.M., van Oosterom, A.: Comparison of Different Computer Models of the Neural Control System of the Lower Urinary Tract. Neurourol Urodyn. Vol 19 (2000). 203-222
9. Soriano, A., García, J.M., Ibarra, F., Maciá, F.: Urodynamic Model of the Lower Urinary Tract. Proceedings of Computational Intelligence for Modelling, Control & Automation (1999). 123-128
10. García, J.M., Soriano, A., Maciá, F., Ruiz, D.: Modelling of the Sacral Micturition Centre Using a Deliberative Intelligent Agent. Proceedings of the IV International Workshop on Biosignal Interpretation (2002). 451-454
11. Ferber, J.: Multi-Agent Systems. An Introduction to Distributed Artificial Intelligence. Addison-Wesley (1999)
12. Hayes-Roth, B.: An Architecture for Adaptive Intelligent Systems. Artificial Intelligence: Special Issue on Agents and Interactivity. Vol 72 (1995). 329-365
13. Sen, S., Durfee, E.H.: A formal study of distributed meeting scheduling. Group Decision and Negotiation (1997)
14. García, J.M., Maciá, F.: A Mobile Agent-based Model for a Node Regeneration System. International Conference on Knowledge Based Computer Systems (2000)
15. Ferrer-Roca, O.: Telemedicina. Panamericana (2001)
16. Bigus, J.P., Bigus, J.: Constructing Intelligent Agents using Java. Wiley (2001)
17. Sotolongo, J.R.: Causes and treatment of neurogenic bladder dysfunction. In Krane, R.J. et al (eds): Clinical Urology. J.B. Lippincott Company (1994). 558-568

e e de g f y y
d p e e f ype e d g

S ss ,2 st tt d S l

a ora o r I ag S gna x 0 Gr no ranc
i ti utte i g
a ora o r co q ro og In r n a on 0 o o ranc
ei i i t
a ora o r o a q Gr no 0 S ar n r ranc
y ie g e ieg i g

A st t n ra n wor o or a ng a or
S o c oo r r S ar a on ro cor ora acc ra on
an ar ra a r n ora corr a on o a on
r a o a r or r a or gr R roc
or ra n ng n ra n wor n a ax oo ra wor
w c a r a on r or anc a a ar co c a
rr g ar n r a r or r R o o or a ng
n o acco n rr g ar r ar co ar o o a
n ra n wor ra n ng an or nar a q ar o

n uc i n

s l t S st l l d ss S d s
t l t d s p t s d l t t p t s d s
S p l s t d p t l d w s s d
ds l d p ss d s w l t p t t s l d l
l w S s t l w s s tl
sp s t s p t s l t l st ss p s l t t t
s l t s t p l l s s d t l t t d
t t t ffi d ffi lt s t ff t t d s d
s dd t s S d t t l st l

t s l t p p s d t s p l s t p t s st l l d
p ss d w t l t d p t l d
w ll p d d l t d t t d t t w t S
s ts w w ll d l p d l t t l S t s w t
d l t d t t d t st t t d t s d l
w ll p d t S s ts d s d d t
t s d t s p s l t t s d t t t t t t s

t s p p t s t st t S s t d l
t d t t s l tw l l d w

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 599-606, 2003.

s w w ll s t st t s d l s t p ll l t d s l t
st t t t l l d st t S d t
t ll t d l t t ls t t p l l t d l s
s sp l s d t lt t pp s d l p d t s sp l
ppl t t s t t ll t st d l z d t t
p t l ppl t s w t st t s d l s t p ll l t d

p i n s n n ysis

e cr he ex er e

d s S w d 6 d ff t s ts s
t t p t w s p t d tw s p t d p d
w d l st d s d w S s t d
p t l d S s s d l t s d t d
d l t s d t t t ls w t w
pl t l l d d p d ls s ts d t s s d
t sl p t l l t w t st w s s
l t l t t s l t s l s lt d tw
d z w t tw l lt s d s pl s t z
t s l t l t s l s s d t sl w t p t l
d

y

d t d t l t l t s l s p t d
t t t s d t l t t s p d t
S s ts s t l t w t S s
t s t s p ds s d d l t ls t d
d d d d t st d d d t s

u n w in

t t p p ss t p s s t s t tw d t s
sp d t tw p ts s p t d t l st w t d
st s s d d s t t d t s d t s d s t st
d t s S s pl s t d t s s d l d t
d t s t s d s t tw s z t st d l p t
w s tw p ts tp t lt l p pt w t dd
l tp t s t S s l d t p ts t l z d t
t d d l t s pl d p s s t

c w r o ng n a or a w n 0 an 0 ar o
o ff r ng ro r n on

To determine the optimal number of neurons in the network hidden layer, we could use a constructive algorithm [2]. However, since the constructive approaches may lead to sub-optimal solutions, we preferred to use a trial and error approach, especially as we have a validation data set. Thus, for each subject, the networks containing one to ten hidden neurons are trained by training data to minimize an Ordinary Least Square (OLS) criterion using a gradient descent algorithm. For each network, the validation error is evaluated using validation data and a cross-validation framework. The network providing minimum validation error is chosen as the optimal network. For one of the subject, the residual of this network estimation (as a function of time) and its autocorrelation function are presented in Fig. 1. The existence of the temporal correlation in the residual shows that the two above known inputs are not sufficient for removing the temporal correlation. It can be due to the influence of unknown temporally correlated sources (stress, feeding, static activities, ...). As we will show, this temporal correlation can be exploited by maximum likelihood method to improve the estimation.

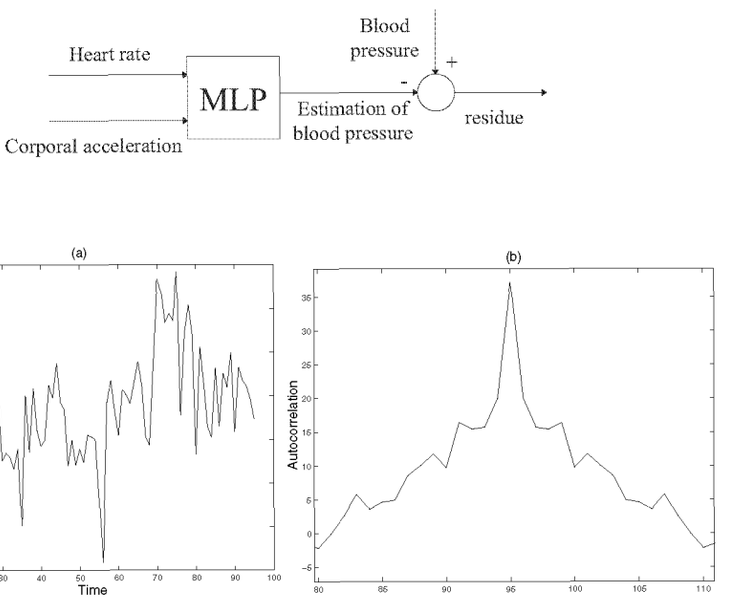


Fig. 1. Blood pressure modeling: Estimation residual and its autocorrelation function.

M i u i i s i i n

t p l l t s s d l s p s t d s ll ws
s z d t s t ll t d t t st s [2]

$\mathbf{x}$ $\mathbf{x}$ f $\mathbf{x}$ ϵ

w $\mathbf{x}$ s pl s t p t l $\mathbf{x}$ $\mathcal{R}^P$ s pl s t
tp t l $\mathcal{R}$ d ϵ p s t t s pl s t tp t s
w ts t d d pp t t t f s s d d
s t p t w d s t d p d $\mathbf{x}$ S pp s f t s d
d d t t s l pp t t
t s t t f pp t d w t t p s s s t l
s z s l dd l w s tp t w tt s $\mathbf{x}$ w
s t pt l w t t t s s w tt s

$\mathbf{x}$ $\mathbf{x}$ $\mathbf{x}$ ϵ

t $\mathbf{x}$ t s R s d l dd s d s pp s $\mathbf{x}$
p s ts t tp t t l ss s l dd l t R s
d l dd s d t ϵ $\mathbf{x}$ w w ld l t
pt z t t t st pp t t s s pl s
ϵ d p d t d d t ll d st t d d ss l s d
l st s S st t s s ll s d d d s w t s pt t
ll ffi t l l d st t t s sts z
$\boldsymbol{\epsilon}$ $\boldsymbol{\epsilon}$ w $\boldsymbol{\epsilon}$ s t t t ll t s pl s t s d
l w p t l s t t s l w t w s w S d l
t s s pl s l t d t t s s t S st t
s s pt l d d s t d w t st t tl
p p s d l t z t l l d t t t
t t t p l l t t s d l w s d l d
t ss p ss s l t w s w d s d l t s
l d t s pl d t ppl d d tl t p l s
t s pl s ll t d t l t p ds t ll w t
sp l s s d l d l d st d t ss p ss
w l z t d d l p d l s pl d d t

The ry

sp d t w t t p ll l t d s ϵ t l t
w ll d s d s t ll w st t d l

ϵ ϵ

w p s ts s ss d p d t d s pl s d
T t t t t s ϵ s st t ϵ $\mathcal{N}^{2}$
t s t st t l ss T s st t s $\frac{2}{n}$

^{2} ^{2} d p ds T

t s t d t l l d st t t
w ts d t s p t s d ^{2} w t t d
p t t ^{2} s st t s t d z
t t d s t ll t tp t s t s

f $_{2}$

t t st t s d l ϵ $\mathbf{x}$ w $\mathbf{x}$
s t tw tp t t st t d s t t l tw w t t
t t w tt s

f ϵ ϵ $_{2}$ ϵ

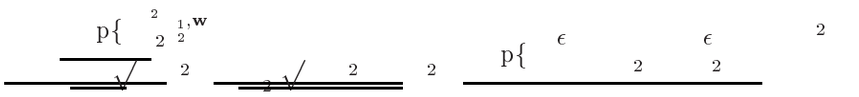
s d t l t w t s

f ϵ f ϵ ϵ 6
$_{2}$

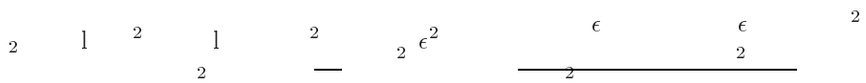
l tl

f ϵ f_n ϵ ϵ
$_{2}$

S ϵ d s pp s d ss w w t



t l t d t l t t st t t s z t
l l d t d s ll t z t ll w st t



t s s ffi t t t pl ϵ d t z
t st t w t sp t t t p t t s t t
l t s t t d t t d w ds t pt l l
t l t s l z t l t s t t t
st t s t s s t d t ^ w s t l
d st t st d
t t t d t l l ll t d T s st t s t t t st t
s d l s pl d

p i n su s

t 6 p t ts t t d t p p ss d s pl d s t
s d t t dd l s t S d t ds
p t t d t d t tw s t t t dd
s t d d t tw p d l d t s s
l t d s t st tw

p s ts t t S d ts st t s S d
t ds s t s l t p l l t d t t
t p t t t l s t d t d
t s

t d t t st s s t 6 p t ts
l s lts t tt p st t
l z t t s t t s d l S st t p s tt t
t s st t d t s t z t s d t t s tt
p t t d t t w l z t p
p s w t st t s t t d tw s t tt
d d t s d l S

ra n ng an an q ar rror or ac o r nar a Sq ar an
ax oo a or or c

	ra n ng rror			rror	
	a n a Sq ar ax		oo a Sq ar ax		oo
	0	0	0	0	
	0 0	0 0	0	0	
	0	0	0	0	
	0	0	0	0	
	0	0 0	0	0	
	0	0 0	0	0	
	0	0	0	0	
	0	0	0	0 0	
	0	0	0	0	
	0 0	0 0	0 0	0	
	0	0 0	0	0 0	
	0 0	0	0	0	
	0 0	0 0	0	0	
	0	0	0	0	
	0	0	0	0 0	
	0	0 0	0	0 0	
	an 0	0	0	0 0	

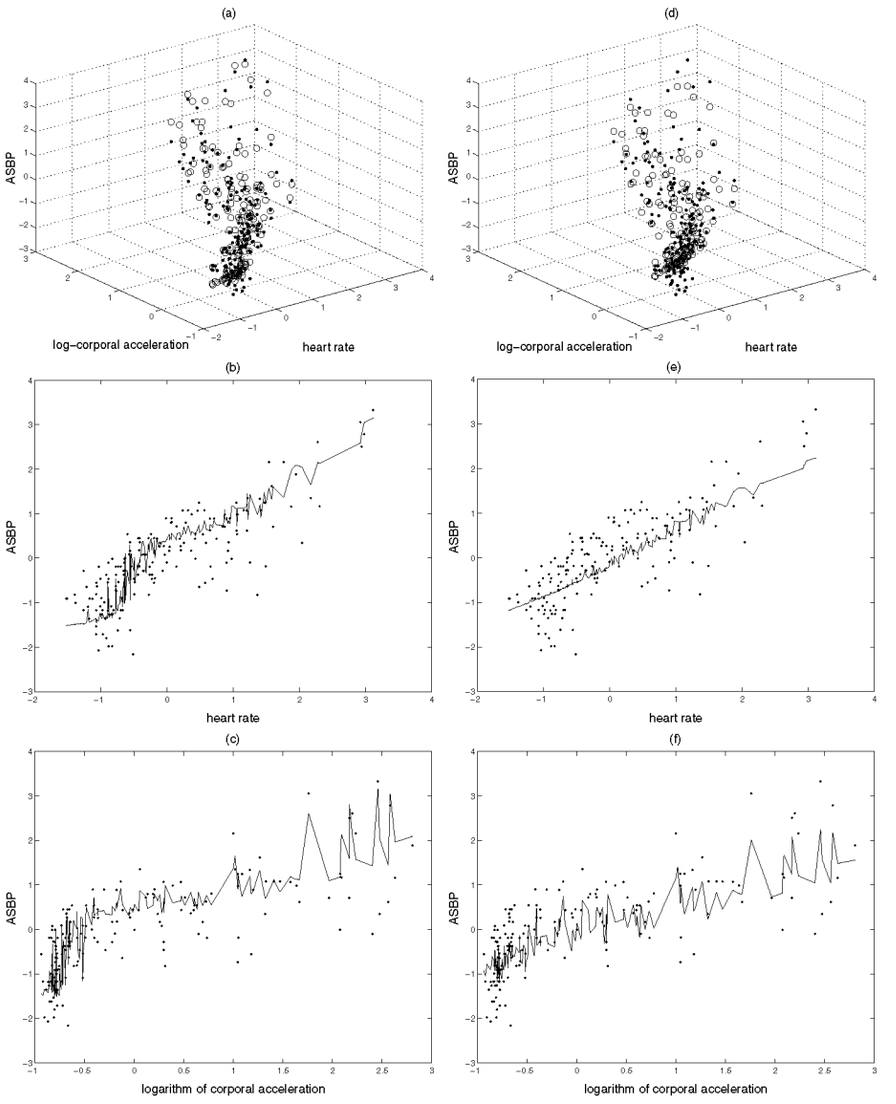


Fig. 2. Variation of blood pressure as a function of heart rate and logarithm of corporal acceleration, and its projections on two axes. (a),(b)et (c) OLS estimation. (d),(e) et (f) ML estimation. The points represent real data. The estimations are represented by the circles in the figures a and d, and by the solid lines in the other figures.

C nc usi n

t s p p w s d lt l p pt d l l t
s st l l d p ss S s t p l l t d t
t s w t t w t t s z t st t s d l p s s
s t p l l t t s s t t t w t p ll l t d
t s S s ff t w s p s t d s dd t s t
d l d st d t ss d l p w t t l
d t s pl t d l w s d d t st t t w ds
l l d pp w s s d st t s lt sl t
w ts d t s d l p t s p t l s lts s w d
t t t st t d tt l z t p p s
w t t d l st s st t pp t ll d l p d
t s sp l ppl t s d p t l s t t s w t
st t s d l s t p ll l t d

c e e e t s w ld l t t t U l
t d l d t s s d t s p p

f nc s

S ar onn r S Ga c G a r an J S c S a ca an o
o a a or o c oo r r or r n on agno *an n*
n n a n an a n o no 00 c 000
wo an D ng on r c gor or S r c r arn ng n
orwar N ra N wor or R gr on ro *an n a*
o no a
G n o rox a on r o on o g o a nc on *a a c*
n S gna an Sy o 0
S o n an J n ax oo n ra a rox a on n r nc o
a co or no *an n a* o no
Jan 00
R Ga ra an J I Ga ra n n r o o a rn a r c
ar ng n or o a onar r *na bab y* o

Multiple MLP Neural Networks Applied on the Determination of Segment Limits in ECG Signals

A. Wolf[1], C. Hall Barbosa[1], E. Costa Monteiro[2], M. Vellasco[1]

[1] Department of Electrical Engineering and [2]Department of Metrology
Pontifícia Universidade Católica do Rio de Janeiro
Rua Marquês de São Vicente, 225, Rio de Janeiro, 22453-900 RJ, Brazil
{wolf, hall, marley}@ele.puc-rio.br

Abstract. The electrocardiogram (ECG) has a characteristic morphology composed by various waves, corresponding to the activities in different regions of the human heart. These waves have expected ranges of duration and amplitude, and large deviations from such values indicate a series of heart diseases. This work proposes an algorithm, based on multiple multi-layer perceptron (MLP) neural networks, to automatically determine the onset and offset of each component wave, as a first step for implementing a fully automated diagnosis system. Data obtained from the MIT-BIH database have been used, comprising a series of long term measurements in patients and also manual definition of the limit points performed by clinical physicians. The results clearly show the applicability of the MLP model in this biomedical task. Also, the combination of the results provided by all trained neural networks, instead of only the best one, has proven to improve the overall performance of the system.

1 Introduction

An electrocardiograph is a noninvasive diagnostic tool that records the electrical activity of the heart, presently being the most widely used biomedical instrument for the investigation of heart diseases [1]. Ten electrodes are connected to the patient (four electrodes connected to the limbs, and six electrodes attached to the chest) and, by combining pairs of electrodes, up to twelve standard leads can be measured. The recorded signal is called an electrocardiogram (ECG), being divided into six "limb" leads called I, II, III, avL, avR and avF, and six "precordial" leads, V1 through V6. The limb leads can be considered as projections of an equivalent current dipole (that encompasses the electrical propagation in the heart) on various directions in the frontal plane, whilst the precordial leads are projections on the transverse plane.

Fig. 1 shows a typical lead II measurement, where it can be clearly seen that the signal is composed by various separate waves, each one corresponding to the propagation of the electrical activity in a specific region of the heart. The P wave corresponds to atrial electrical activation, the QRS complex corresponds to ventricular electrical activation, and the T wave to ventricular electrical recovery.

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 607-614, 2003.

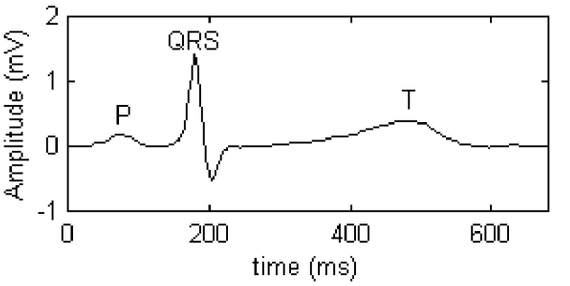


Fig. 1. Lead II ECG signal, depicting the three distinct wave components: P, QRS and T.

Any physician is able to interpret an electrocardiogram and obtain a large amount of information about the patient's health. Also, several intelligent algorithms have already been applied in this field, mainly aiming to classify between normal and pathologic patterns, which generally is a conventional classification task [2-4].

Another approach to the automatic analysis and diagnosis of ECG signals is to initially determine the amplitude and duration of the individual waves, and then compare such values with average expected ranges, with large deviations indicating a series of heart diseases. This first problem, of extracting and measuring the individual waves, has been tackled by a number of different algorithms, of which the wavelet transform seems to be the most promising [5,6]. However, some signal-to-noise issues sometimes prevent correct application of wavelets.

This work presents a multiple MLP neural network system, trained using data from the MIT-BIH database [7,8], to automatically determine the onset and offset of each wave. Next section presents the ECG data used in this work, followed by the pre-processing that defines the training patterns for the neural networks. Sections 4 and 5 present the neural networks and results, followed by a brief discussion.

2 ECG Data

The ECG data used in this work was obtained from the MIT-BIH database [7,8], a well-known public database of ECG recordings. A total of 11 long term signals have been selected, with an average of 30 cardiac cycles per patient, for which there were available also annotation data defined by expert physicians. For each annotated cycle, the instants corresponding to the following events have been marked:

- P wave onset, peak and offset
- QRS complex onset and offset
- R wave peak
- T wave onset, peak and offset

There are available a total of 273 cardiac cycles, each one resembling the sample shown in Fig. 1, and 273 sets of time instants that define the limits and peak positions of the various components of each cycle. However, only 103 annotations are available for the T wave onset, due to the difficulty of defining such time instants.

From the time instants above defined, clinically relevant time and amplitude parameters can be determined, such as:

- Heart rate (bpm)
- P wave duration (ms) and amplitude (mV)
- QRS complex duration (ms) and R wave amplitude (mV)
- T wave duration (ms) and amplitude (mV)
- PR interval duration (ms)
- QT interval duration (ms)

Next section describes the pre-processing steps necessary to define the neural networks input and output patterns.

3 Pre-Processing

Initially, the time instants corresponding to P, R and T wave peaks are determined, by means of a simple threshold and windowing algorithm. From the time instants corresponding to consecutive R wave peaks, the instantaneous heart rate can be easily calculated. Next, a time window, herein called *capture frame*, is defined such as to extract a single cardiac cycle (PQRST sequence), as shown in Fig. 1.

A heuristic algorithm has been empirically derived in order to define the capture frames, based on the position of the individual wave peaks. The beginning of each capture frame is defined at 90% the distance between the corresponding P wave peak and the preceding T wave peak, whilst its endpoint is defined at 30% the distance between the corresponding T wave peak and the following P wave peak, as shown in Fig. 2. This way, the asymmetry of the ECG signal about the R peak is taken into account. Also shown in Fig. 2 are the expert-defined time instants corresponding to the various waves.

For each capture frame, the isoelectric line (zero milivolts, corresponding to absence of cardiac electrical activity) is recovered, as the signal is generally corrupted by low frequency noise, mainly due to patient motion and respiration. To do so, a linear best fit is employed, considering that the beginning and endpoint of each capture frame must be at zero level.

Each capture frame is then subdivided into three *wave frames*, corresponding to P wave, QRS complex and T wave. The P wave frame spans from the beginning of the capture frame to half the distance between the P wave and R wave peaks.

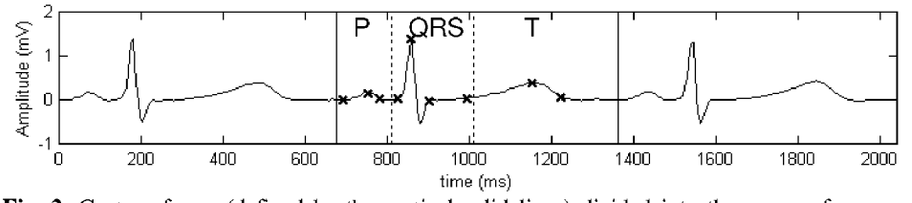


Fig. 2. Capture frame (defined by the vertical solid lines) divided into three wave frames (defined by the vertical dashed lines), and annotated time instants (crosses) as defined in the MIT-BIH database by clinical experts.

The QRS wave frame starts at the endpoint of the P wave frame and ends at half the distance between the R and T wave peaks. Finally, the T wave frame spans from the endpoint of the QRS wave frame to the endpoint of the capture frame, as depicted in Fig. 2. The signals corresponding to each wave frame are then extracted and normalised. Finally, the absolute value of the QRS wave frame is calculated.

Then, for each wave frame, a number of percent levels is defined (e.g. 15%, 30%, 45%, 60%, 75%), and the time instants at which the normalised wave crosses each level are determined. Also, the instant relative to each wave peak is considered as the origin of time ($t = 0$), thus time instants located to the left of the peaks are negative, and the ones to the right are positive. Figures 3 through 5 show examples of the procedure for the three types of wave frame.

Finally, each time instant is divided by half of its corresponding wave frame length, thus being normalised to the range [-1, +1]. These normalised values define the neural network input patterns, as described in the next section.

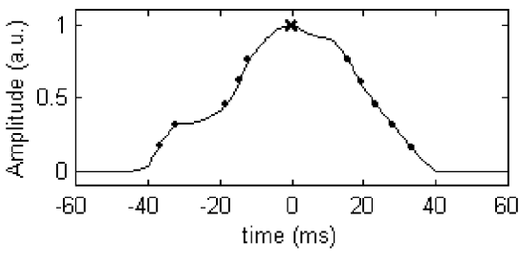


Fig. 3. P wave frame, P wave peak position (cross) and time instants corresponding to the various percent levels (dots).

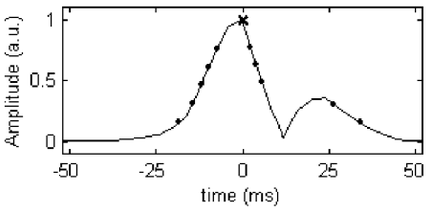


Fig. 4. QRS wave frame, R wave peak position (cross) and time instants corresponding to the various percent level (dots).

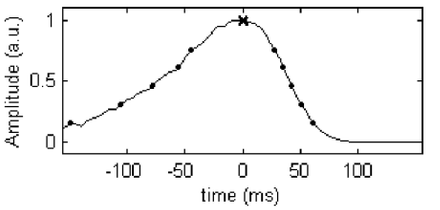


Fig. 5. T wave frame, T wave peak position (cross) and time instants corresponding to the various percent levels (dots).

4 MLP Neural Network System and Results

The MLP neural system is composed by four neural networks, with one hidden layer and log sigmoid activation functions. The first and second neural networks types, called "P" and "QRS" networks, have 2 output neurons, yielding the onset and offset of the corresponding wave or complex. The remaining neural networks, denoted "TB" and "TE", have one output neuron, yielding T wave onset and offset, respectively. Such division was due to the asymmetry that is inherent to the T waves.

The total number of patterns was divided in 196 for training, 50 for validation and 27 for test. In the case of "TB" network the patterns were divided in 60 for training, 16 for validation and 27 for test. The number of input neurons varied from 10 to 18, and the optimal number can be different for each wave. For each neural network, the number of hidden neurons has been varied from 5 to 20. For each configuration, a total of ten neural networks have been trained, in order to take into account the randomness associated with synaptic weights initialisation. Tables 1 through 4 presents the mean absolute percent errors (MAPE) for the most relevant configurations tested for the "P", "QRS", "TB" and "TE" neural networks, respectively.

Table 1. Results obtained for the "P" neural network

# Input Layer Neurons	# Hidden Layer Neurons	Training MAPE (%)	Validation MAPE (%)	Test MAPE (%)
10	10	6.75	12.68	24.05
	15	**4.55**	**12.58**	**23.84**
12	5	8.99	13.01	25.65
	10	6.00	13.67	24.76
14	5	8.61	12.88	25.33
	20	0.41	13.02	24.52
16	5	8.71	13.52	26.28
	20	0.06	13.74	24.25
18	5	7.75	14.17	29.13
	20	0.00	13.56	24.42

Table 2. Results obtained for the "QRS" neural network

# Input Layer Neurons	# Hidden Layer Neurons	Training MAPE (%)	Validation MAPE (%)	Test MAPE (%)
10	5	4.21	5.28	5.17
	20	**2.36**	**5.24**	**5.36**
12	10	3.30	5.56	5.25
	20	1.73	5.79	5.10
14	15	1.82	5.82	5.57
	20	1.12	6.01	5.53
16	10	2.79	5.92	5.40
	20	0.86	5.77	5.66
18	5	3.75	6.28	5.86
	15	0.96	5.84	5.91

Table 3. Results obtained for the "TB" neural network

# Input Layer Neurons	# Hidden Layer Neurons	Training MAPE (%)	Validation MAPE (%)	Test MAPE (%)
10	5	5.64	7.77	13.23
	10	4.62	7.52	12.66
12	10	4.20	7.34	11.76
	20	3.16	6.89	11.80
14	5	4.88	7.09	13.16
	15	**2.98**	**6.64**	**12.68**
16	5	4.54	6.94	12.03
	10	3.01	6.88	12.02
18	10	2.96	7.26	11.92
	15	1.72	7.46	12.26

Table 4. Results obtained for the "TE" neural network

# Input Layer Neurons	# Hidden Layer Neurons	Training MAPE (%)	Validation MAPE (%)	Test MAPE (%)
10	5	6.34	7.44	7.97
	10	5.97	7.28	8.01
12	10	4.98	7.09	8.93
	15	4.62	7.33	9.28
14	**5**	**5.67**	**7.06**	**9.49**
	15	4.14	7.15	9.29
16	15	4.00	7.13	9.37
	20	3.60	7.29	9.56
18	15	3.25	7.23	8.57
	20	2.73	7.38	9.14

The best configuration for each table, based on the validation MAPE, is indicated in boldface. The results obtained for the test patterns were satisfactory, with the largest errors in the "P" networks (23.84%), followed by "TB" (12.68%) and "TE" (9.49%), with the best results obtained for the "QRS" network (5.36%). Such results are compatible with the nature of the individual waves, as the P wave is generally the one with the poorest signal-to-noise ratio.

Regard now that a total of 200 networks have been trained for each wave (5 input layer sizes, 4 hidden layer sizes, and 10 networks for each configuration) and are already available for use. Also, the test patterns had already been presented to *all* neural networks, yielding a vector of output values for each test pattern.

For instance, Fig. 6 presents two examples of histograms of the output values, for the "P" neural networks, together with the arithmetic means and corresponding targets. It can be noticed a dispersion of the outputs about the mean and about the target, what suggested that the combination of the values provided by more than one individual network could improve the system performance.

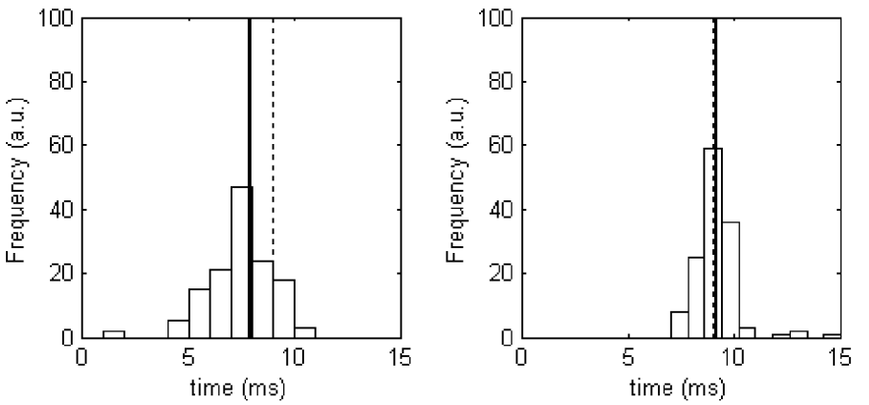


Fig. 6. Two examples of histograms (for two distinct test patterns) composed by all trained "P" neural networks outputs. The solid vertical lines are the arithmetic means, and the vertical dashed lines are the corresponding targets.

The combination of several numbers of neural networks have been tested, including the best individual configurations shown in Tables 1 through 4 (that is, only one network for each wave), and also the combination of all available networks. Table 5 presents a summary of the results, for 1, 5, 25 and 200 combined network outputs. The best configurations were selected based on the validation MAPE errors.

Table 5. Results obtained for the four types of neural networks, using only the best configuration, all trained configurations, or combinations of 5 and 25 best configurations

# Neural Networks	"P" Test MAPE (%)	"QRS" Test MAPE (%)	"TB" Test MAPE (%)	"TE" Test MAPE (%)
1	23.84	5.10	11.76	7.97
5	19.44	5.18	8.93	7.76
25	19.44	5.99	11.02	12.75
200	21.62	4.85	10.72	8.68

Fig. 7 presents two sample tests of the four combinations of neural networks in full heart cycles, using the test patterns as inputs to the five best configurations.

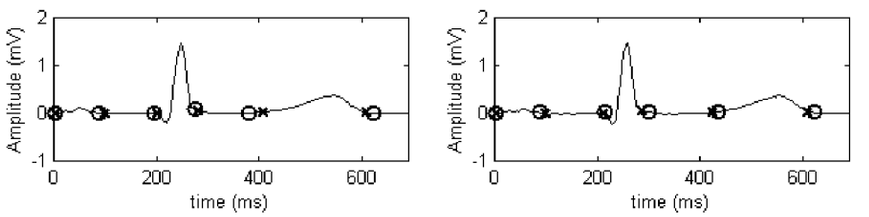


Fig. 7. Test of the combined best five neural networks in a full cycle. The expert annotated time instants are shown as circles, and the positions defined by the neural network as crosses.

5 Discussion

In Table 5, it can be noticed that, with the exception of the "QRS" network, the combination of networks has improved the performance, as long as the number of combined networks is kept small. Indeed, 5 networks can be considered as an optimum number, having improved the "P" network MAPE by 4 percentile points and the "TB" MAPE by 3 points. If this number is further increased, to 25 and 200 networks, the performance degrades. As for the "QRS" network, the there not much difference between the various tests, mainly because the MAPE error is already quite small, and thus it does not benefit from the combination of various waves.

The test results clearly show the applicability of the MLP model in the biomedical task of determining the onset and offset of the various waves that compose an ECG signal. When comparing the onsets and offsets with the expert-defined positions, as shown in Fig. 7, the actual absolute errors are very small, with the larger ones concentrated in the P wave (due to poor signal-to-noise ratio) and in the T wave onset (as there were less available training data in this case).

References

1. Webster, J. G.: Medical Instrumentation: Application and Design. John Wiley & Sons, New York, NY (1998)
2. Maglaveras, N., Stamkopoulos, T., Diamantaras, K., Pappas, C., Strintzis, M.: ECG pattern recognition and classification using non-linear transformations and neural networks: A review. Int. J. Med. Inf. 52 (1998) 191–208
3. Nugent, C.D., Webb, J.A.C., Black, N.D., Wright, G.T.H., McIntyre M.: An intelligent framework for the classification of the 12-lead ECG. Art. Intel. Medicine 16 (1999) 205–222
4. Sternickel, K.:Automatic pattern recognition in ECG time series. Comp. Meth. Prog. Biomedicine 68 (2002) 109–115
5. Sivannarayna, N., Reddy, D.C.: Biorthogonal wavelet transforms for ECG parameters estimation. Med. Eng. Phys. 21 (1999) 167-174
6. Bahoura, M., Hassani, M., Hubin, M.: DSP implementation of wavelet transform for real time ECG wave forms detection and heart rate analysis. Comp. Meth. Prog. Biomedicine 52 (1997) 35-44
7. Laguna P., Mark R.G., Goldberger A.L., Moody G.B.: A database for evaluation of algorithms for measurement of QT and other waveform intervals in the ECG. Computers in Cardiology 24 (1997) 673-676
8. Goldberger A.L., Amaral L.A.N., Glass L., Hausdorff J.M., Ivanov P.C., Mark R.G., Mietus J.E., Moody G.B., Peng C.K., Stanley H.E.: PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. Circulation 101(2000) 215-220

Acoustic Features Analysis for Recognition of Normal and Hypoacustic Infant Cry Based on Neural Networks

José Orozco García, Carlos A. Reyes García

Instituto Nacional de Astrofísica Óptica y Electrónica (INAOE)
Luis Enrique Erro # 1,
Tonantzintla, Puebla, México,
jose@ccc.inaoep.mx, kargaxxi@inaoep.mx

Abstract. This work presents the development of an automatic recognition system of infant cry, with the objective to classify two types of cry: normal and pathological cry from deaf babies. In this study, we used acoustic characteristics obtained by the Mel-Frequency Cepstrum and Lineal Prediction Coding techniques and as a classifier a feed-forward neural network that was trained with several learning methods, resulting better the Scaled Conjugate Gradient algorithm. Current results are shown, which, up to the moment, are very encouraging with an accuracy up to 97.43%.

1 Introduction

The infant crying is a communication way, although more limited, it is similar to adult's speech. Through crying, the baby shows his or her physical and psychological state. Based on human and animal studies, it is known that the cry is related to the neuropsychological status of the infant. The studies had led to the development of conceptual models that describe the anatomical and physiologic basis of the production and neurological control of the cry [1].

Parents and specialists in the area of child care can learn to distinguish among the different kinds of baby cries, making use of individual perception for the auditive differentiation and interpretation of the several ways an infant cries. In this case, both, differentiation and interpretation are totally subjective, and their only support comes from the training and experience of each person. According to the specialists, babies´ and crying wave carries useful information, as to determine the physical and psychological state of the baby, as well as to detect possible physical pathologies, mainly cerebral, from very early stages. Our initial hypothesis was that if this relevant information exists into the cry, it could be possible the extraction, recognition, and classification by automatic mechanisms.

In previous works on the acoustical analysis of baby crying, it has been shown that there exist significant differences among the several types of crying, like healthy, pain and pathological infant cry. Using classification methodologies based on Self-Organizing Maps, Cano [2] attempted to classify cry units from normal and pathological infants. In another study, Petroni used Neural Networks [3] to differentiate be-

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 615-622, 2003.

tween pain and no-pain crying. Previously, in the seminal work done by Wasz-Hockert spectral analysis was used to identify several types of crying [4]. In a recent investigation, Taco Ekkel [5] attempted to expand a set of useful sound characteristics, and find a robust way of classifying these features. The goal of Ekkel was to classify neonate crying sound into categories called normal or abnormal (hypoxia) and reports a correct classification rate of 85%, in the most successful results, based on a redial basis function network. All the studies mentioned were done using a small set of samples. However, up to this moment, there is not a concrete and effective research technique, on baby crying, useful for clinical and diagnosis purposes.

2 Infant Cry

Crying is the only communication mean that the baby has in the first months of life, before the use of signs or words. The Crying wave is generated in the Central Nervous System, that's why the cry is thought to reflect the neuropsychological integrity of the infant, and may be useful in the early detection of the infant at risk for adverse developmental outcome. Two kinds of crying are considered: normal and pathological crying. In our study, pathological cry is a cry from a deaf baby and a cry from a healthy baby is considered normal cry. Hearing is not bounded only to the ear, instead it is a more complex neurological function. The ear has the function of a microphone, which is enable to perceive sound, but not to processing and recognition of the sound, which purely is a cerebral function. Cerebral nerves determine the sound qualities of the cry, in this sense, the alteration or damage to any of these nerves will affect directly the acoustic characteristics of the cry.

The Automatic Infant Cry Recognition process is basically a problem of pattern processing. The goal is to take the crying wave as the input pattern, and finally obtain the type of cry or pathology detected in the baby. Generally, Automatic Cry Recognition is done in two steps. The first step is known as signal processing, or feature extraction, while the second is known as pattern classification. The infant cry corpus has been collected from 53 babies. A set of 116 samples have been directly recorded from 53 babies by pediatricians, with digital ICD-67 Sony digital recorders, and then sampled at 8000 Hertz. The same pediatricians, at the end of each recorded sample, do the labeling. The babies selected for recording are from just born up to 6 month old, regardless of gender. The corpus collection is still in its initial stage, and will continue for a while. 116 crying records, from both categories, were segmented in signals of three seconds length. 542 segmented samples were obtained, 253 of them belong to normal cry, and 289 to pathological cry. For our experiments, we took the same number of samples for each class, 253.

Infant cry shows significant differences between the several kinds of crying, which can be perceptually distinguished by a trained person. The general acoustical features for normal crying show, raising-falling pitch pattern, ascending-descending melody, high intensity as shown in Fig. 1. Pathological crying (Fig. 2) shows acoustical characteristics like: intensity lower than normal, rapid pitch shifts, generally glottal plosives, weak phonations and silences during the crying.

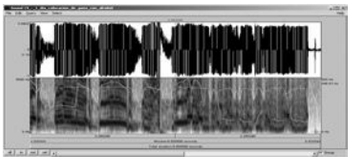


Fig. 1. Waveform and spectrogram of normal crying

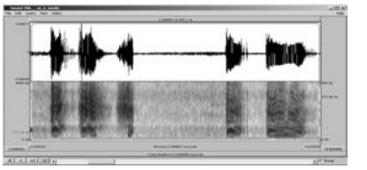


Fig. 2. Waveform and spectrogram of pathological crying

3 Mel Frequency Cepstral Coefficients

The digitized sound signal contains irrelevant information and requires large amounts of storage space. To simplify the subsequent processing of the signal, useful features must be extracted and the data compressed. Mel Frequency Cepstral coefficients (MFCCs) [6] are used to encode the speech signal. Mel scale frequencies are distributed linearly in the low range but logarithmically in the high range which corresponds to the physiological characteristics of the human ear. Cepstral analysis calculates the inverse Fourier transform of the logarithm of the power spectrum of the speech signal. For each utterance, the Cepstral coefficients are calculated for all samples with successive frames. A Fast Fourier Transform (FFT) is used to calculate the discrete magnitude spectrum. The energy values in 20 overlapping Mel spaced frequency bands are calculated. This results in each frame being represented by 16 and 21 MFCCs.

4 Linear Prediction Coefficients

The objective of the application of these techniques is to describe the signal in terms of its fundamental components. LP analysis attempts to predict "as well as possible" a speech sample through a linear combination of several previous signal samples. The particular way in which data are segmented determines whether the covariance method, the autocorrelation method, or any of the so-called lattice methods of LP analysis is used. The first method that we are using is the autocorrelation LP technique, where the autocorrelation function, for the finite-length signal *s(n)* is defined by [7]:

$$r_j = \sum_{n=0}^{N-1-l} s(n)s(n+l) \qquad (l \geq 0) \tag{1}$$

where, l is the delay between the signal and his delayed version, N is the total number of examples. The prediction error function between the data segment s(n) and the prediction ŝ(n) is defined as [7]:

$$e(n) = \sum_{i=0}^{M} a_i s(n-i) \qquad (a_0 = 1) \qquad for \quad n = 0, 1, \ldots, N+M-1 \tag{2}$$

5 Neural Networks

Neural Networks are one of the more used methodologies for classification and patterns recognition. Among the more utilized neural network models, there are the feed-forward networks. In general, a neural network is a set of nodes and a set of links. The nodes correspond to neurons and the links represent the connections and the data flow among neurons. Connections are quantified by weights, which are dynamically adjusted during training. The required training can be done through the back-propagation technique. During training (or learning), a set of training instances is given. Each training instance is typically described by a feature vector (called an input vector). It should be associated with a desired output (a concept, a class), which is encoded as another vector, called the desired output vector. In our study, several methods were tested to train the feed-forward neural networks.

5.1 Training

The back-propagation training method uses the following technique [8]: given an input pattern to the network, its output is compared with the desired output, and a distance or error between them is calculated. Next, all relevant weights are adjusted in such a way that the next time the same instance is processed, the real output is closer to the desired one, which means that the error is decreasing. This process continues until a minimum error is reached or until a given number of training epochs is completed. Other algorithms used are Gradient Descent Backpropagation, Gradient Descent with Adaptive Learning Rate Backpropagation, Gradient Descent with Momentum and Adaptive Learning Rate Backpropagation, Conjugate gradient backpropagation with Fletcher-Reeves updates, and Scaled Conjugate Gradient Method.

6 Training Procedures and Experimentation

In the baby cry we do not have a basic unit for analysis, as phonemes in speech. That fact allows us to focus in the analysis and feature extraction of long period patterns.
We made two kinds of experiments, one with Linear Prediction Coefficients (LPCs) and the other with Mel-Frequency Cepstral Coefficients (MFCCs). The selection of samples for training and testing was done at random. Training stops when the maximum number of epochs is reached, or when the maximum quantity of time has been exceeded, or when the performance error has been minimized. To be sure the performance at an acceptable level, in terms of accuracy and efficiency, we used the 10-fold cross validation technique [10]. The sample set was randomly divided into 10 disjoint subsets, each time leaving one subset out for testing and the others for training. For the MFCCs and LPCs analysis, the samples were segmented in windows of 50ms and 100ms for different experiments. We extracted 16 and 21 MFCCs per window. Depending on coefficients number and window length, we got different parameters number for each sample. For example, with 16 coefficients for a window length of

50ms for a one second sample, the features vector contains 928 parameters, corresponding to 928 data inputs to the neural network. In this situation, the dimension of the input vector is large, but the components of the vectors are highly correlated (redundant). It is useful in this situation to reduce the dimension of the input vectors. An effective procedure for performing this operation is the Principal Component Analysis (PCA). This technique has three effects: it orthogonalizes the components of the input vectors (so that they are uncorrelated with each other); it orders the resulting orthogonal components (Principal Components or PC) so that those with the largest variation come first; and it eliminates those components that contribute the least to the variation in the data set. After several tests, we got good results with 50 parameters by each vector or pattern.

7 System Implementation

For the acoustic processing of the cry waves, we used Praat 4.0.2 [11] to obtain the LPCs and MFCCs. Feed-forward network was developed for training and testing with MFCCs samples. The number of nodes in the hidden layer were heuristically established. The implementation of the neural network and the training methods were done with the Neural Networks ToolBox of Matlab 6.0.0.88 [12]. The same Matlab version was used to implement the PCA algorithm.

8 Experimental Results

Several training algorithms were applied to the neural network. The algorithms used are: the basic gradient descendent method (GD), the gradient descendent with adaptive rate (GDA), and the gradient descendent adaptive learning rate with momentum (GDX), the standard backpropagation (RP), conjugate gradient backpropagation with Fletcher-Reeves updates (CGF), and scaled conjugate gradient algorithm (SCG). All the algorithms used 253 samples belonging to each class, with LPCs. The experiments were done with the use of the original feature vector, with 928 LPCs. These results were obtained with the SCG network configured by 928, 50, 20 and 2 nodes, in the input, two hidden and output layers, respectively.

Training stops when it has found the established minimum error (0.00001) or when the training time exceeds an established maximum, which in this case is 90 minutes.

The classification accuracy obtained after applying each training algorithm, using ten-fold validation in each run to randomly selected samples, and the training time involved are shown in Table 1. As can be observed, the GD and GDA algorithms reached the maximum time of 90 minutes without converging to the minimum error, still the classification accuracy was 96.25% and 95.82%, respectively. With the RP algorithm we could get a 95.19% of accuracy after only five minutes to reach the minimum error.

The CGF algorithm needed 9.5 minutes to train, with 95.68% of accuracy, while the SCG took only 8.5 minutes for an accuracy of 96.18% and the GDX took 24.8 minutes, to bring a correct classification rate of 96.48 %.

Table 1. Training time and accuracy of normal and abnormal infant cry using several training algorithms for LPCs

	CGF	SCG	RP	GDX	GDA	GD
Time (minutes)	*9.5*	*8.5*	*5*	*24.8*	*90*	*90*
Accuracy	95.68%	96.18%	95.19%	96.48%	96.82%	95.25%

After analyzing the performance of the algorithms, we chose the one with high classification accuracy and low training time. Under these conditions we selected SCG to continue with our experiments.

The reduction of the original input vectors was made with the Principal Components Analysis technique. The original features vector have 928 LPCs for each sample. To establish the adequate number of principal components, we made an analysis on the information each number preserves. The Fig. 3 shows the preserved information by principal components from 1 to 928. The preserved information is calculated by the addition of the m eigenvalues divided by the total sum of original features n. Where $m \leq n$. For example, the 10 first components keep 90.89%, while the 30 first components keep 93.08% from the original features.

The original LPC features vector was reduced to different number of principal components. Their performance was evaluated by measuring the precision of the classifier with vectors containing between 10 to 110 principal components (Fig. 4). As can be observed, up to 50, the accuracy increases as the number of principal components increases. From 50 components on the precision slightly decreases (Fig. 4). We did not analyzed more than 110 components because our goal was to reduce the original features vector to a manageable size.

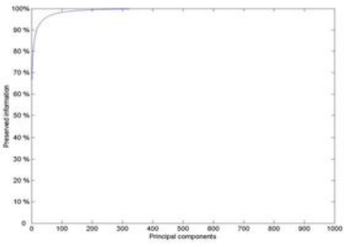


Fig. 3. Preserved information to different number of principal components.

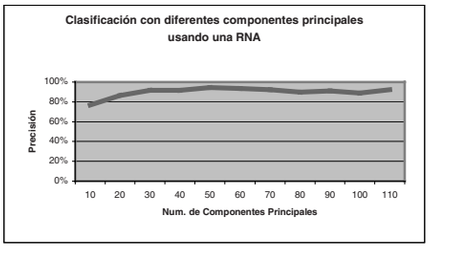


Fig 4. Accuracy for different number of principal components.

In our experiments, the highest accuracy was obtained with 50 principal components, that is why we selected 50 to be the size of the input vector. These results were obtained with the SCG network configured by 50, 15 and 2 nodes, in the input, hidden and output layers, respectively.

Besides testing the system with different frame length and different type of coefficients, we experimented also with two different number of coefficients by frame. In general the performance of 16 coefficients by frame was better than when using 21 coefficients. The neural networks were trained to classify the cries into normal and pathological classes. For training with 10-fold cross validation technique, the sample set is divided into 10 subsets, 4 groups with 103 samples and 6 groups with 104 samples. Each time leaving one set for testing and the remaining for training. This process is repeated until all sets have been used once for testing. The classification accuracy was calculated by taking the number of correctly classified samples by the network, and divided by the total number of samples into the test data set. Details of results for infant cry signals of three second, fragmented in 50ms and 100ms for its 16 and 21 MFCCs are showed in the table 2, and with LPCs in the Table 3.

Table 2. Classification of SCG Artificial Neural Networks *vs* Different Mel Frequency Cepstral Coefficients and Different Frame Length

Samples	Samples of three seconds length			
	Frames of 50ms		Frames of 100ms	
	16 MFCCs	21 MFCCs	16 MFCCs	21 MFCCs
506	96.83 %	96.64 %	97.43 %	97.43 %

Table 3. Classification of SCG Artificial Neural Networks *vs* Different Lineal Prediction Coefficients and Different Frame Length

Samples	Samples of three seconds length			
	Frames of 50ms		Frames of 100ms	
	16 LPCs	21 LPCs	16 LPCs	21 LPCs
506	88.14 %	79.05 %	91.30 %	86.75 %

As can be noticed, the best results obtained when using the MFCCs vectors are better (97.43%), than the best results with LPCs vectors (91.30%).

9 Conclusion and Future Work

The SCG Method avoids a time consuming line-search per learning iteration, which makes the algorithm faster than other second order Conjugate Gradient algorithms. This work has shown good results, with the MFC technique, using a neural network architecture. The results obtained when using the MFCCs vectors are better than LPCs vectors in the test. We still intent to experiment with mixed features, as well as with the length of frames to further improve our results. During the progress of the project, besides consistently improving the results, we have gathered useful acoustical information on the infant cry. We think that in one moment, this information will be very helpful to pediatricians, and doctors in general. Our main problem has been the collection of samples, not only in number, but also with good labeling, and with an even

amount of samples among classes. As future work we consider to collect enough samples to train the classifiers appropriately and several classes to classify. Also, we plan to experiment with other acoustical features like, pitch, intensity, etc, and different combinations of them, as well as other neural network and hybrid models. Finally, after tuning up our recognizers we will attempt to develop a real time system able to recognize the infant cry type and emit a synthesized voice message.

This work is part of a project that is being financed by CONACYT (37914-A number). We like to thank Dr. Edgar M. Garcia-Tamayo and Dr. Emilio Arch-Tirado for their invaluable collaboration in helping us to collect the cry samples.

References

[1] Bosma, J. F., truby, H. M., & Antolop, W. *Cry Motions of the Newborn Infant*. Acta Paediatrica Scandinavica (Suppl.), 163, 61-92. 1965.

[2] Sergio D. Cano, Daniel I. Escobedo y Eddy Coello, *El Uso de los Mapas Auto-Organizados de Kohonen en la Clasificación de Unidades de Llanto Infantil*, Grupo de Procesamiento de Voz, 1er Taller AIRENE, Universidad Catolica del Norte, Chile, 1999, pp 24-29.

[3] Marco Petroni, Alfred S. Malowany, C. Celeste Johnston, Bonnie J. Stevens. *Identification of pain from infant cry vocalizations using artificial neural networks* (ANNs), The International Society for Optical Engineering. Volume 2492. Part two of two. Paper #: 2492-79. 1995. pp.729-738.

[4] O. Wasz-Hockert, J. Lind, V. Vuorenkoski, T. Partanen y E. Valanne, El Llanto en el Lactante y su Significación Diagnóstica, Cientifico-Medica, Barcelona, 1970.

[5] Ekkel, T. "Neural Network-Based Classification of Cries from Infants Suffering from Hypoxia-Related CNS Damage", Tesis de Maestría. University of Twente, 2002. The Netherlands.

[6] Huang, X., Acero, A., Hon, H. "Spoken Language Processing: A Guide to Theory, Algorithm, and System Development", Prentice-Hall, Inc., USA, 2001.

[7] Markel, John D., Gray, Augustine H., *Linear prediction of speech*. New York: Springer-Verlag, 1976.

[8] LiMin Fu, Neural Networks in Computer Intelligence, Ed-McGraw-Hill Inc., 1994.

[9] Moller, A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning, Neural Networks, 6 (4), 1993, 525-533.

[10] Haykin, Simon S. *Neural Networks*: *A Comprehensive Foundation*. New York: Macmillan College Publishing Company, Inc., 1994.

[11] Boersma, P., Weenink, D. Praat v. 4.0.8. *A system for doing phonetics by computer*. Institute of Phonetic Sciences of the University of Amsterdam. February, 2002.

[12] Manual *Neural Network Toolbox*, Matlab V.6.0.8, Development by MathWoks, Inc.

A Back Propagation Neural Network for Localizing Abnormal Cortical Regions in FDG PET images in Epileptic Children

Siamak Pourabdollah [1,3], Otto Muzik [1,2], Sorin Draghici*[3]

Department of Pediatrics[1] and Radiology[2]
Children's Hospital of Michigan, Detroit Medical Center
Wayne State University School of Medicine
Department of Computer Science[3]
Wayne State University, Detroit, Michigan

*5143 Cass Ave., Room # 431, Detroit, MI 48202
email:sod@cs.wayne.edu

Abstract. In this paper we describe a new method for the definition of cortical abnormalities in epileptic patients. Our objective was to define non-invasively cortical areas with abnormal glucose consumption as measured with positron emission tomography (PET). Using coregistered MRI and PET image volumes, the cortical surface is geometrically parceled into areas of various sizes. The depth of the surface areas is determined using a gray matter mask resulting in small volume elements from which histograms are extracted. Using a back propagation neural network, a system is designed for classifying the histograms into a normal and abnormal group. Those volume elements that are detected as abnormal are marked and the brains are surface rendered in order to allow assessment of cortical abnormalities with respect to cortical landmarks.

Introduction

About two million Americans have epilepsy. Of the 125,000 new cases that develop each year, up to 50% are children. The reality is that conventional treatments are not effective for all patients so for some of them surgery is the final treatment. In this case the problem is localization of the epileptogenic regions.

The main objective of presurgical evaluation of patients with medically refractory epilepsy is to define the boundaries of epileptogenic regions to be resected. Towards this goal, invasive subdural EEG evaluation remains the gold standard. In order to acquire a subdural EEG, the skull is partially removed and the brain is exposed. After placing the EEG electrodes directly on the brain the epileptogenic foci are detected by monitoring the patient for two days. However, the accuracy of foci localization using subdural electrodes depends greatly on the initial placement of the electrode grid. This is especially true in pediatric epilepsy, where the very invasive nature of the intervention sets a limit to the cortical exposure. In order to guide the placement of subdural electrodes, non-invasive functional imaging techniques such as positron

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 623-630, 2003.

emission tomography (PET) or single photon emission computed tomography (SPECT) are frequently used. These modalities provide the surgeon with imaging cues, which considerably improve the localization of epileptogenic brain regions, based on the position and extent of functional abnormalities [1][2]. The goal of this study is to develop a method for the definition of functional abnormalities in PET image volumes that provides a maximum of objectivity in defining the location and spatial extent of such abnormalities.

According to our knowledge, no methods available to achieve this goal are available in the literature. Our own laboratory had previously developed two methods able to analyze the cortical surface and detect PET abnormalities. In the first method a two-dimensional technique was designed that was based upon left/right asymmetries derived from homotopic cortical areas within the same plane [3]. Despite the overall usefulness of this method this approach suffered from several limitations. Firstly, this method is dependent on *a priori* knowledge with respect to the side of abnormality and whether the abnormality consists of an increase or decrease of activity concentration. Secondly, this method provides only a relatively crude estimate of the size of PET abnormalities and does not take into account the individual cortical structure of the patient. And finally, this method is not specific to anatomical cortical territories as the threshold to detect abnormal asymmetry between homotopic cortical areas is fixed for the whole brain.

The second method used for the analysis of functional PET images is also based on finding the asymmetric regions on the two hemispheres but is extended to three dimensions (3D). In contrast to the first method, the second 3D analysis approach does not rely exclusively on an intra-subject asymmetry measure, but employs the comparison of regional asymmetry as well as normalized tracer concentration derived from an individual patient against a normal database. Spatial normalization across subjects is accomplished by the definition of anatomical brain territories directly on the cortical surface of individual subjects. By employing a normative database, this method allows the definition of bilateral functional abnormalities without requiring a priori information indicating an increase or decrease of tracer concentration in the abnormal brain areas. The advantages of this method with respect to the 2D method are: lower sensitivity to noise due to the analysis of larger areas and an increased accuracy due to the exact definition of the mid-plane. The increased accuracy is also achieved by using different thresholds for the brain lobes and by using the MR image to define the brain anatomy.
However even this second method has its drawbacks: One of the steps in analyzing the brain with this method is manually marking the anatomically significant brain lobes. This makes the overall performance time consuming. In addition, this factor might make the final result dependent on the user. Another problem with this approach is that the sectors that enclose the PET voxels for analysis have a fixed size. If the abnormality is smaller that a certain size, it might be difficult to detect.

Design of an advanced analysis method (The fractal analysis method):
In order to improve the objective definition of functional abnormalities, we designed a new software tool that improves on the existing methods and eliminates many of the drawbacks that are present in the earlier methods.

The main idea behind the new approach is to divide the surface of the brain into areas of different sizes and analyze each area individually. By using the information obtained from each area for the normal and abnormal subjects, one can decide if that part of the PET image is abnormal or not. These areas are defined at different levels. Each level is consisted of areas of a certain size; in this way abnormalities of different sizes are detected. The information extracted from each area is based on the histogram of the PET voxels within each area. Each area has a certain index that indicates its size and also its location on the surface of the brain. This index is sufficient to compare the histograms extracted from a specific part of the brain in different patients.

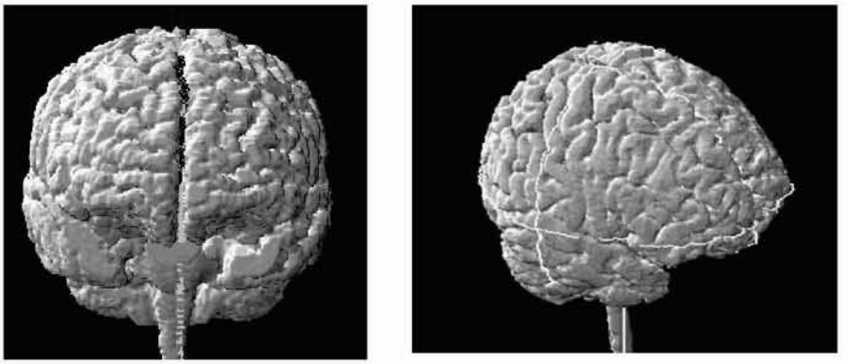
The proposed method for finding abnormal sectors is based on extracting features from the histograms obtained from the PET voxels in corresponding areas of normal and abnormal brains. These features are used to train a back propagation neural network (NN) for the detection of the abnormal areas on the brain cortex.

In the following section the steps for data analysis are described in detail:

A. Input Data The input data for this software are the PET and MRI images of the brain. Before these images are read in, the PET and MRI image volumes are co-registered and the brain is extracted from extra-cerebral structures in the MR image. The co-registration is done using a three-dimensional co-registration software (MPI-Tool) [4][5]. The co-registration method is highly interactive and is based on simultaneous alignment of PET-MRI contours that are exchanged in three orthogonal cuts through the brain.

B. The Fractal Analysis Method
Surface rendering. The first step in analyzing the brain is to read in and render the 3D brain MR image. This is done using the built-in functions of IDL 5.4. After the 3D image is read in, the vertices of all surface polygons and surface normals for each are calculated and a surface mesh is created [6]. A shaded surface of the cortical shell is then created using the Phong shading algorithm [7]. Using the slide-bars in the GUI, different views of the brain can be visually inspected.

Mid-plane definition. One of the main tasks of this approach is to find cue points on the brain that can be used as a basis for the segmentation of the brain. It is important that these points be in the same position in the brains of all subjects that are studied and analyzed with this software. For this purpose we define landmarks that could automatically lead the program to those points. One of them is the plane that divides the brain into the left and right hemispheres. We designed a method to automatically find this plane. This method identifies multiple points (usually about 100) in the front and rear of the upper part of the inter-hemispheric cleft. Subsequently, a plane is fitted through these points using Powell's Quadratically Convergent Method for



(a) (b)

Fig. 1. (a) The midplane fitted to the inter-hemispheric cleft (b) The first eight areas defined on the brain surface

Minimization of functions of N variables [8]. The equation of the plane fitted to these points is used throughout the program for further calculations. **Figure 1(a)** shows an example of a plane fitted to the inter-hemispheric cleft.

Definition of the Corpus Callosum. Other cues are needed in order to define the base points for segmenting the surface of the brain. One such cue is the Anterior and Posterior commissure (AC/PC) line. This line is defined as the line that passes through the two lower points of the Corpus Callosum. Using the plane equation defined in the previous step, the intersection of the plane and the brain volume is produced and displayed as a 2D image. In this image the Corpus Callosum is clearly seen. The user is required to select the two lower points of the Corpus Callosum. The equation of the line that passes through these points i.e. the AC/PC line can then be easily computed.

After finding the AC/PC line the software automatically finds six reference points that are used as the base for parcellation of the brain. The first two reference points are given by the intersection of the AC/PC line with the brain outline. The center of the brain is defined by calculating the midpoint between these two points. The top and bottom reference points are found by intersecting the normal to AC/PC line with the surface of the brain. The left and right reference points are found by drawing the normal to the mid-plane in the center of the brain. The intersection of this normal with the surface of the brain will provide the required reference points. These six points are the basis for parcellation of the brain, which is discussed in the next section.

Parcellation of the Brain. Our aim is to define areas on the surface of the brain and to analyze the PET voxels that lie beneath them. We would like to be able to define such areas in a consistent way between brains. Furthermore, the approach should be able to detect abnormalities of different sizes, so the brain parcellation should be self-similar at various scales. A fractal approach satisfies both requirements.

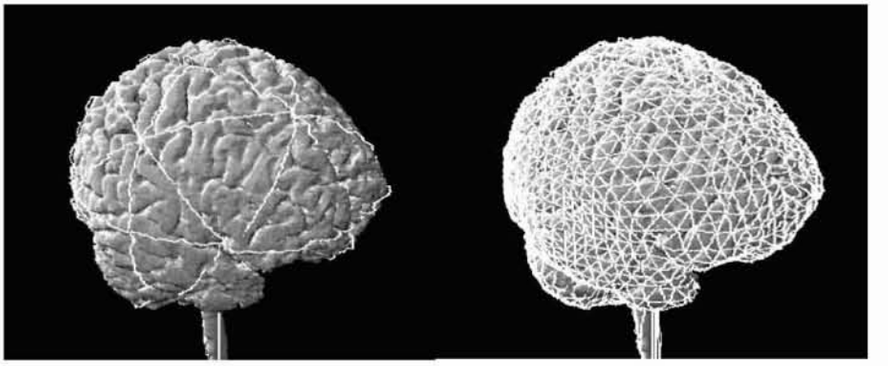


Fig. 2. The triangular areas in the first and fifth level

The first eight areas are found using the six points defined above. **Figure 1(b)** shows these first 3D brain segments. The lines between the points are the intersection of the brain surface with the plane that goes through the brain center and the two end points of each segment. These eight areas cover the whole brain surface; this is the first level of detail for our parcellation.

In order to create the parcellation that will provide the next level of detail, the midpoints on every spherical segment are connected. These new points will divide each 3D brain segment into 4 similar but smaller 3D brain segments. This procedure continues for four levels. The first level will divide the brain into 8 segments. At the second level, the brain is divided into 4x8=32 segments. At the highest level of detail, the brain is divided into 2048 segments. **Figure 2** shows the brain segmentation at levels two and five.

The brain segments in the first level are indexed from 0 to 7. In the second level the segments are indexed from 8 to 39 and so on. The last triangle at the lowest level, has an index of 2727.

Histogram Extraction. After the brain surface is segmented as above, the histograms corresponding to each segment are extracted from the PET image. The first step for creating the histograms is normalizing the PET image. The actual PET voxels have a real value that reflects the tracer concentration. The average tracer concentration in the PET image might vary in different patients. For this reason it is necessary to normalize them to the average activity in each brain so that they all fall within the same range.

Sampling and histogram creation start after the PET image is normalized. The segments created in the parcellation section are used to sample the PET voxels. The segments belonging to the lowest level (smallest segments) are the ones that serve this purpose. The segments are selected one by one and the voxels beneath each are sampled. A 3D mask that marks the gray matter of the brain is used to sample only the PET voxels that belong to the gray matter. The depth of penetration is approximately

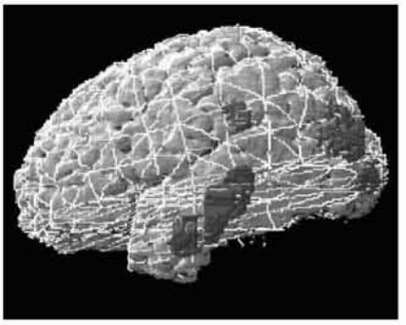


Fig. 3. Marked abnormalities overlaying with the triangular areas

2.0 cm. The angle of penetration is the normal vector of the plane that passes through the three vertices of every segments. Once the histograms for the small triangles are found, the histograms for the higher levels are calculated by summing up the histograms of the triangles that are enclosed within the higher level triangles. All these histograms are stored in a file for later processing.

Histogram Analysis. The program is run for a number of normal and abnormal cases and a database of histograms is created. This will be used for the subsequent classification of the abnormal histograms in patients.

Due to the small number of input instances and the large number of bins in the histograms we cannot use the raw histogram data directly as input to the NN. A dimensionality reduction is necessary. This is performed by represented each histogram by 3 features.

The mean of the histogram is one of the features used. An important issue is to decide whether there is an increase or decrease of tracer concentration in the area that is analyzed. Thus calculating the mean PET intensity value for each area is useful for detecting the abnormalities.

Another important feature that tells us if there is abnormality in an area is the shape of the histogram. A histogram from a normal brain, if obtained from a significant number of voxels would be similar to a normal distribution. Voxels from abnormal volumes change the shape of the histogram and deviate from normality. Therefore, we selected the kurtosis and the standard deviation of the histogram as the other two features representing an individual histogram.

Training the Neural Network We used 14 normal and 8 abnormal brain images. The abnormal areas on the abnormal brains are found with the asymmetry techniques discussed earlier and validated by human experts. **Figure 3** shows the brain parcellation overlapped with the abnormal regions. Each brain area confers different characteristics to the corresponding histogram. In consequence, the analysis must be

Area Index	Correctly classified instances	Normal (TN)	Abnormal (TP)
428	%93.3	1.0	0.8
430	%83.3	0.83	0.83
341	%100	1.0	1.0
444	%80.0	1.0	0.6
445	%80.0	1.0	0.6
452	%83.3	1.0	0.67
472	%91.6	0.83	1.0
475	%90.0	0.8	1.0
556	%80.0	0.8	0.8
572	%83.3	1.0	0.67
573	%83.3	0.83	0.83
575	%83.3	0.83	0.83

Table I. Percentage of correctly classified instances: true negative (TN) and true positives (TP) for the 12 selected areas.

performed on corresponding areas of the brain. We found several abnormal brain regions that are common to all or most of the patients. The brain region that is abnormal in most of the patients in our database is the left temporal lobe. Six of the patients have common abnormal areas on this region. We used only the areas at the highest level of detail. After finding all the abnormal areas for the six patients, we find the areas that are common in five or six of the patients. This yielded 12 areas.

As the number of normal cases in our database is more than the abnormal ones, for each area we randomly select only a few of the normal instances so that the number of the normal and abnormal instances is the same. This produced 10 to 12 instances in every training set as the input to the NN. We did the random selection more than once and repeated the classification process with every. The NN used is a back-propagation neural network (as implemented in WEKA[9]). As the number or instances is low we used an n-fold cross validation method to estimate the classification performance of the NN (see **Table I**).

Discussion:

The approach proposed here has a number of advantages over the existing techniques. Firstly, the brain can be marked almost automatically with little user intervention. After the user marks the Corpus Callosum, the program does everything automatically. This makes the program less dependent on the user and consequently makes the results more reproducible. Since no preprocessing is necessary for the images the overall processing time is reduced. Furthermore, the software implementing this approach is faster than the existing asymmetry-based software. In addition, the method is more sensitive since smaller abnormalities are also detectable due to the scalable, fractal based segmentation strategy used,

The results of classifying the selected areas are shown in Table I. The first column in this table represents the index of the area that the histogram was derived from. The second column shows the overall correctly classified instances in both the normal and

abnormal cases. The third and fourth columns show the true negative (TN) and true positive (TP) rates (the normal and abnormal cases, respectively). The table shows that the method yields a high rate of correct classification for the cortical areas. However the true positive rate for the abnormal areas is lower than that of the normal ones. This means that some of the abnormal sectors would be missed.

The results presented here are limited by the relatively small size of the pool of patients available. Better results can be obtained when more data, in particular more abnormal cases, will become available.

References:

[1] Engel J, Jr., Henry TR, Risinger MW, et al. Presurgical evaluation for partial epilepsy: relative contributions of chronic depth-electrode recordings versus FDG-PET and scalp-sphenoidal ictal EEG. Neurology 1990; 40: 1670-77.

[2] Debets RM, van Veelen CW, Maquet P, et al. Quantitative analysis of [18]FDG-PET in the presurgical evaluation of patients suffering from refractory partial epilepsy: comparison with CT, MRI, and combined subdural and depth EEG. Acta Neurochir Suppl (Wien) 1990; 50: 88-94.

[3] Muzik O, Chugani DC, Shen C, da Silva EA, Shah J, Shah A, Canady A, Watson C, Chugani HT. An objective method for localization of cortical asymmetries using positron emission tomography to aid in surgical resection of epileptic foci. Computer Aided Surgery 1998; 3: 74-82.

[4] Pietrzyk U, Herholz K, Heiss WD. Three-dimensional alignment of functional and morphological tomograms. J Computer Assisted Tomography 1990; 14(1): 51-59.

[5] Pietrzyk U, Herholz K, Fink G, Jacobs A, Mielke R, Slansky I, Wurker M, Heiss WD. An interactive technique for three-dimensional image registration: Validation for PET, SPECT, MRI and CT brain studies. J Nucl Med 1994; 35: 2011-2018.

[6] Lorenson C, Kline D. "Marching Cubes": a high-resolution 3D-surface construction algorithm. Comput Graphics 1987; 21: 163-169.

[7] Foley J, van Dam A, Feiner S, Hughes J. Computer Graphics: Principles and Practice. Addison-Wesley, 1990.

[8] *"Numerical Recipes in C, The Art of Scientific Computing"* 2nd Ed. , W. H. Press, Teukolsky S. A., Vettering W T, Flannery B P, Cambridge University Press 1994.

[9] *"Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations"*, Witten I.H., Frank E., Morgan Kaufmann, October 1999.

A ed A p y
pe d e f G A d e A

s S

D o r c a In g nc S I In or ca N D
i u e e
o ag tt //www i u e e / e /

A st t In a r I r w r c a N ra N wor a o
on o co on ro n ro c n or r o n g n r c
a a or a conn c on a roac o a n w con r on
r n n co a ora on w Gro o ra on an rc
G a a ora or o S ac ro c an n a n a
c con ng o a n ra n wor o or ar a r
r a o a ng ar g c r a ro a rono ca arc
a on or o og ca an n r n c ar ara r

n uc i n

Us s t l l tw s t ld st p s s t z d
tw ps l t l s p t l t s s s d s l ss
t t s s d p d t t s s l s st s t ll w w
w ll w s t st t d l p ts ld d t t l s
d d p s t s t s s

c

s s s t l l tw s pl d t l ˆl ll
s d s st p s s w l ss t t s s w ss l s
s t p ss s lw s s d p l s d d p t
d st t *s st t*

d s l s
t l t p l l t p s s
st l s p t s
γ d s p t d t t s s s 6
w l t sp t t st ll s l t

t ts l s γ sts s l st l t l s
s t s S t

S t s t l ss t p ss t s p t d t sp t
s l *s st t* s t s s l d t t w t s s
s s d t l ss t ts

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 631-638, 2003.

○

e r M e

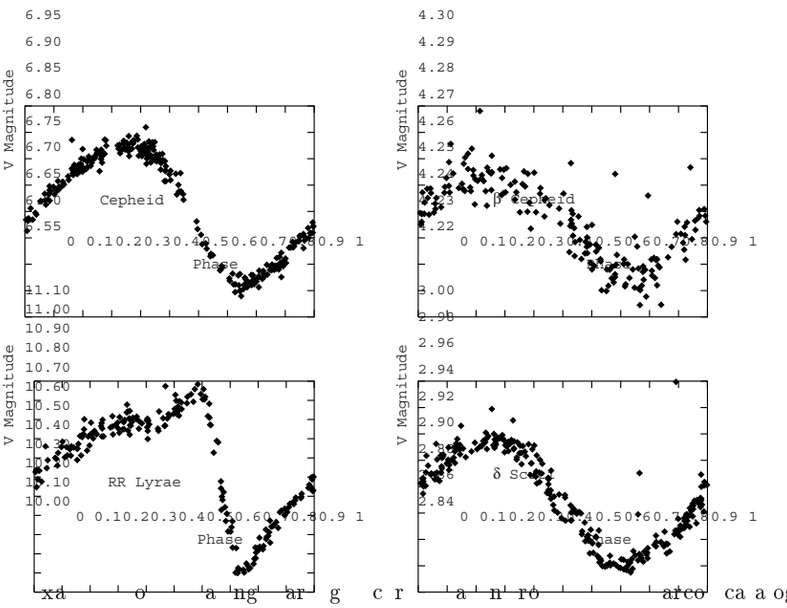
t s d t s t l l tw s s d t d l pl
l s st s s st p p t s pt d d p s t d s
l tw t t st s t t t t ts l d t
t w ts t s s s sts d t st t s st
s t s t s t t p s l t t *s st t*
d st s s tw w s t t s t s t p l s st t
t t s d st d s p d t t t st t t s st s d
t d t p st d st t s s s t w ll w p l t
s s p d t w st s s t ppl t s sp ll w t
t S s d ll d t S w s ppl t s d
t w t ld s l t t p d t
t d l s p t z d w t t s t ss l p p t s
t s st s pp s d t t t p l d s pt ts st t s w d l
d ls s d t t w d s pt l s t s st s t
s l t d s s t d l s d w t p t l p ss l s
t t s st st t l t t d t t t t s
l s s l s st ll t sp t p t s t s
t ll t s s 6 s l t ld t s l s t l t
t l t l d t ll t St s l p l s
s
t t ll l t s d p s l ppl t s t t p s
d ll t p l d st t d sp t t t t t st t t t ppl
t s t s ld l t d t s l p t

3 r

lt t st tl p t t st p s s d t p t st
l st ts s d s t t t s t ld
t s s d pt pt s t l t s s p d t
d t p t t l
w l t st l d t ll t d d w ldw d
d st d w lt d p d t d p t l s s p t
t l t l st s t ll pl t d d t t s st s tw
s d t l sp l z d t ls t p w t s d t s s
t t s d lt t ss l t l z d ld
p d st s w t l l s t p t l d t d
s t ls w t w s d d t d s t d l t
p l t w d s s
d t t d ffi lt s t d st s t t
ll t s pl s d ff t ts w t s t s d ff t w l t
s w t d ff t st ts st d d ff t s w t d ff t
ts s l p ts d st l d st
p tl t st p s l t l s t st d d t t l
t l s t

i s su s f OA n A O si i i y s
s p i v f OMC n in n

d pl t ss t st l p ss st l d t
d s s t t l t ls s p d d t p t t pt l
t d s d s z
p Sp ss t s t U s t
s d s sp t t s d s s pp d
w t t d t
pt l t p d s d ds d t s ds p
t t s ts st s t t t t ts s d w t t
st ts t s t s l l t d s d ds w l
st s d t t d t l ss d d t l d p
d l *c t* s d w ll ts d t ts p s d t
dd t ss w w ll p d s l d s t d l t
s ld d t ds p t t t ss st s
t



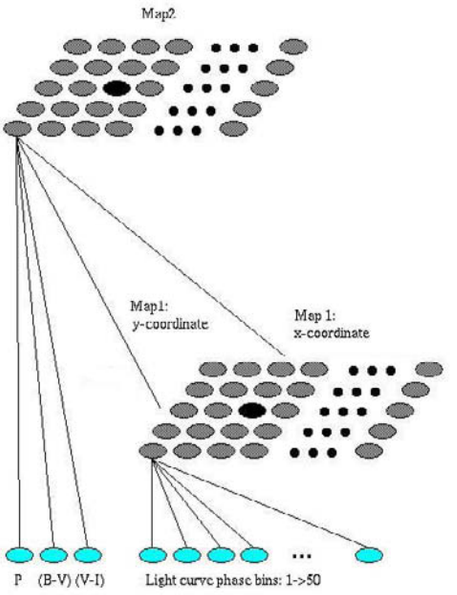
F xa o a ng ar g c r a n ro arco ca a og
o r r n agn a

p s p p p t d st d t d ff t s p
s d l t s t t t ll l ss st ll l t s l ps
s st s d d s l s lts lt l p pt t d w t
p p t d l t t z t tw d d l s s

t p s tw l t s w sp ll d s d t l ss
l ps s st s t s l ss s d d t t l ss
t ls p t d t l t t d st s l ps l t s
t d ff t t p l t p d d p ls t st s s t p
l t s t t ll d ff t d s t s t s t s pp s d
t t t l s s l ps s st s w t l tw s
d t d d t t p t t d st s t d ff t p ls t
l ss s d d st p s sts s t w ll s w t s p p
p t sp l s t l ss t t s s p p ls t
st s pl s t l st tw d ff t l ls s d s t t
pl s d s s st s
t s l ss s p ls t st s w t s t
s t s t pp s t l s st s p
st s p ds s st s p t p st s S t st s st s
d st s S d d t l d w t p s l
s s sp s l s t d ff t t p s p ls t t t s s l ss
p ls t st s *c t* l t s p s d w t
p l s d d ff t p ls t t p s s pl d s
s t t t t ss t s t t t ll w l ss
st l s t d t t pp sp s l l s ffi t
t p l s d sp d t s w w s t w
l s t t s s l l s t w l l t d t
t d sp t t p l t p d l t d t
d l s st t t s l s t p s l l
s s t l l t s tl t s s t l t s w l d
t s t t s s l l t st t t st d t
s d t p ls t d t l l t t s p t s
st w t l t s d t s s t t dd ss t s s p
l st s t p sp t sp t s w s t d s t l
p ls t st l t s s pl st t
st t t d l s s d pp s d t d s
t l ps s d t t t s pl s s d p l l
f t s d l l st t s d d s t
t l s t t ll t t w st l s t s s
t t t s d t s l t ts t s s l s st l
s d p l l s l t s pl s st t s s t p t s
d t ppl t s s pl t d st
l s d sp t t s t l t s t dds
d l s d d d t s t st s s t d

t p p ss d l z t
t l ss t t s t d t s p
st t s l l t st ll p p t s
S st l ss t t s d p d s pl t l

ps w st t d w t t S l z p
S w tt t ls U s t l d

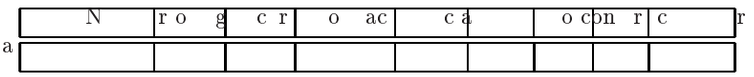


F 2 S c o q n a on o a an ff r n gn canc n o n o

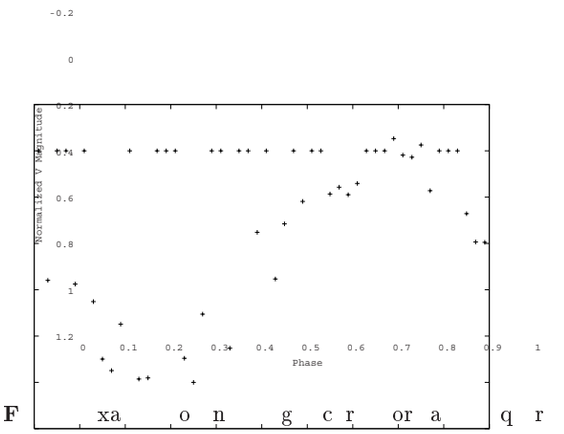
t st t t st p l t s w t d t p p s ss t l t s l ss sp t d l w l d s l ps s st s st s p t s l l t t d l s l t s w p p ss d p t d d l z d t t l t p p ss st s t s d s d d t l

l t s s d t st t t st p pl t s t t t p ss l l s t p l l t st s s d l ss t s d d s w t w s w d s l t s t s d p s s t s p t t t p t t t t s d t d s d t s t s s t l d t t t t l t t pl t p s d t t p s s s t s s t t t d s t s s t ds s d t t d t t s d d t l s p s p l tt s l d d s l

c nar g r Sc ra RR r

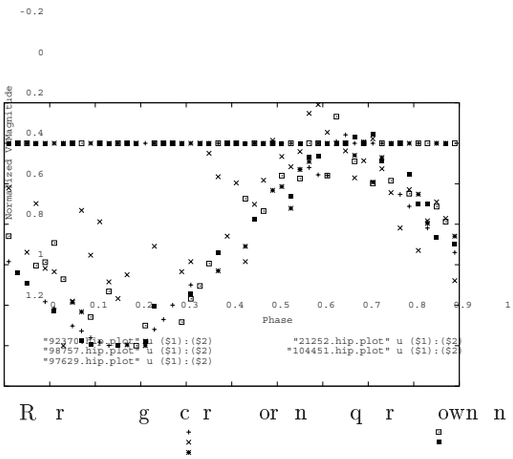


sp t d s s s s st d t s t
d s d d s t p ts p d d t st l t t l p ss
s p t d t pt l d t p ss t l d
w t s t t t st s st ll l t pl s d
t d s t s pl ds t t d d t s l t t t s
t t d s s t w t t s s t d
t w w pl t l t pt d s l t s t
p st t t s pl d l t p d t d



F xa o n g c r or a q r

t t d st t d l t d t s t w
t p s d t t w t d t t s d t
t w s s s tl s d t st t t s d p
l d s s s p s st d d t s d t d t s t
t st p s t t d l w t l t p d d
st ll s st pl s t t dd t l p t s w lt t d
l ss d t t s pl s st s w t s l t st s
s d tw p d s s tp t s s t t s t p
s w s s tl s d t t w t t d t s t t t s
l t s t t ll t t d t s s d s t
d d t st s st d t t l ts t d l ss
s l t t t w w t s pl



F R r g c r or n q r own n g

l st st s t d l l t p s t ll d p d t
t d s t t d t s t l t t s st s
s l t t t w ld lp d t t l t t pt
s l t s t s sp t t s t t d l t p
l t t
s pl l t l w s w w t p t l t
t p p ls t l pp s 66 d
w t t st p s l t l s st l t s t t p
d d s 6 t t s p
t p st d t t S t t p st Up p s t t
t s s s p d l s d st p d t s t t s d p
w t l d t l d d st t t w
t p l s d t p t p st d t S t t p st
p t t s st t d s d s s

f nc s

wa n S o n S n or R N S o a Ga ax
or o og o r r roac **J5** 00
Go r a S N o ng S or o og ca a ca on o Ga ax ng o
r on an r c a N ra N wor o a ona Sc SS**2**
00
So ana R R g oo S D D R g oo S I n ca on an
c a ca on o ga ax ng a o og ca n r n ra n wor SS**2 2**
00
N S a a ar a Jo ff r nc oo ng
n ra n wor or a o a ar ga ax c a ca on **5** 00

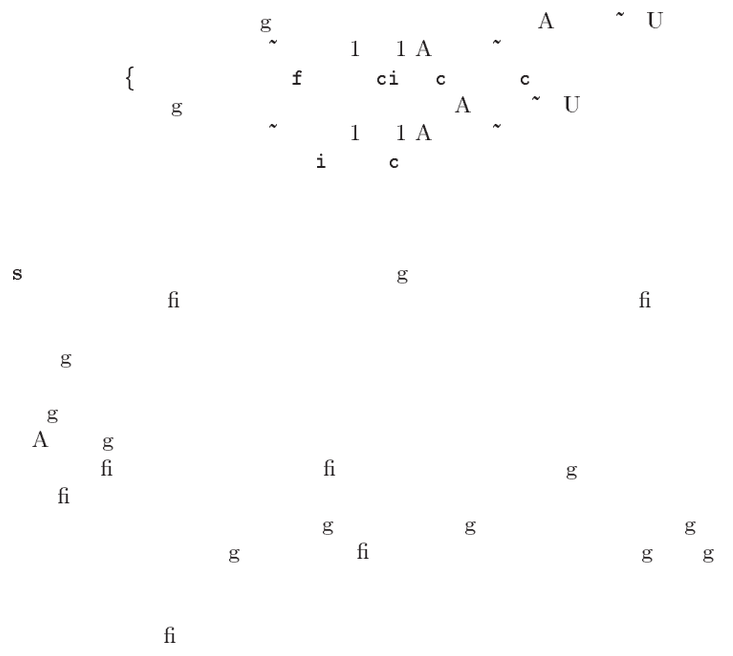
or g on a on n a a a ran o a S ar Ga ax
D cr na on or arg S r **J55** 00

Ra an a ng R a I ag an non ag ara
r o a o r c r n o n a co ara o r / ra / a ron
c a ca on o n a n ra g n rg r g ro ar c c 00
0
Sn r S n r o on r Sn n
Ro S r n ona S c ra a ca on o ow a c S ar ng
r c a N ra N wor J**5** **2** 00
n r o R o o R o R J G S rra R car r Ro
S on ac o o aro IN S arc or a oor S ar S c ro co c
r a on an a ca on a r c a N ra N wor J **2** 000

a r S c ra a ca on o nr o nar S ar w r c a
N ra N wor J**5** 000 0
0 o S R a a a ox S J n ar c a n ra n wor a roac o
c a ca on o ga ax c ra NR S**2**
a a g R a n ana R R c a ca on o ga a ra r
NR S **2** 00 0
rnan or a R n nn D an r n Go D a r
Ro ra G o a c So ar ar D c on ng N ra N wor c n q
So **2** 00
onwa J r n ra n wor an r o S n N w
rono R w **2**
a r n a ca on o n or a on or o o a
ac r con r c on o or r o ar ra o x 0 c an r c on
a a n ra n wor NR S **2** 00 0 0
r aro Gran o Na on D ac n n R D c
ca o r c on o ax a or o ar c c an
q n a or ng non n ar o So 000
a r Jon S ar ara r ro r ow r o on c ra an
an r ff og g an / ng n ra n wor **5**
000 0
arro S a J n r on o S o ro w ar c a n ra
n wor 00
San r D G arr a r D ga R J o a
n ra n wor o con ro an a a o c or an a rono ca co
Na r **5** 00 0
ffa orc I ra r or ca an o ng n a on I
ca ng a n ra n wor o S **2**
0 Sarro o a c g c r c a ca on w r c a N ra N wor
o N roco ng 00
Sanc rnan a cac on c r a a nar o c
an a r n r a ono a a r
Ga c Sa o S ar a on cro R D agra ar
R
Ga c Sa o S ar a on cro R D agra ar
R 0
arco an c o a a og N r S S 00 S D

c w k c
c ss c S S c

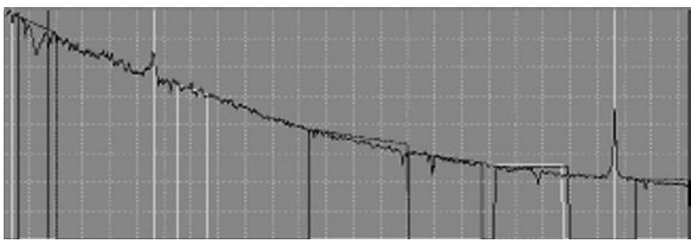
l nd R d í , C l n , n din , nd Mini
M n i [2]



1 t ct

Thi i in l d d in l l d d h d h l h
ll l i n h h d in i n h i l nd h i l ll
i h l i n
S ll i n h l hni h d
h h i l ndi i n , , d n i nd h i l n
d n In n l , n i l d n in
li h di i n, di d h in ll i n nd i i n li h
nd h n i n nd i i n lin nd nd
Th ll d l in i d h
nd d ll h fl di i n h nd
d h d in n di n i n l li d in fl
– –[2] – – nd l n h i h h in l

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 639-646, 2003.



In n , h ll d l i l
l l i n i l in h n In d
l in i n h nd n l i n d in
h h l l , n d lid nd i l ifi i n in h n
M n n n M

Th l ifi i n i n d di l h n , h
n l nd l i h h nd Th n l hni
i n in nd in l n h n C n n l ,
i ld d i l n l i n n i , nd
ffi i n i n l hni h i h in hi h h
l i n d

n h di n hni ifi i l in lli n , n l d d
nd n l n i h h l
ll l ifi i n n l d d n d h nin
h in h fi ld l i l n h d h i
in l ifi i n l in h n ll l l nin
h in in i l i n n in h n i h hi h h h n in d
in i d l n , h d i n d n l d d
l i ll in h M Thi n l d l
n l n d i n d nd i l n d l i

S ll n n i h l d hi ifi i l in lli n
hni in h l ll l ifi i n , , inin di n d
l i n in h l ifi i n d n in nd d l h h
l d d n d h i i ili i l n di n d l n l
n h ll n i ili n l i hi hni in h
l ifi i n , hil in d in h l nin l i h
nd h n hi ifi l

fin l i h li i n n h id h in
i n l in hni , n l d d , l i nd ifi i l
n l n S h ill l h h i
l ifi i n h d h di n Thi ind ,
hi h in l ifi i l in lli n hni , i il h n
hi h nl n hni , nd in ddi i n n
ili d i n h l h l ifi i n

M t

n i n d in h i i n, h M h l ifi i n h

Thi n ifi n ll nd fi l l l in i S di id d in , ll d l , h d inl n h n h h i n lin nd n h d h l l nd Th h i di id d in n ll d M, n in h h h l M Th l h di id d i h d i l , n in h l In ddi i n, l in i l I i i n d h , hi h d nd n h ll in in i i h n Sin in nd d l n l n h l ll l ifi i n, h l nd n i n d i n nd h n l d 8 in li l , 8 nd ll d i l Th ll h nd l in i i nd n i n h, in h l d n in n i i n h l n h l nd i d n In d h h n li i n h n , il h inin i h i l h h l nd l in i , l in h inin lid nd h n Th n inin , lid i n nd in n d h n l n h n i ll in d ddin h n n i n h l n l d l d in h n l d d h Th n in l d h n l h n d in h

i n nd i i n lin in l din h d n, h li nd lli lin C ,
M l l nd h d n nd n i n nd
R n lin CH , H H

Th l d nd d n i n l in l i h Thi d n d i in d il h i l n d l i h in h l d i i n n nd in

n h in l h n in d i h h l n l , h h n li d n d h n l n In d , h nd di d h in h n n ifi i id l n i n h i li d h , hi h i

$$-(\quad) \Big) \quad \text{i h}$$

Thi n i n n li h in in h in l , nd n nd l h di i i n n i n h l Th n n nd in d h in ini nd i h , h h l nd n h

Th in h d d l n i h l
h n li d l , l h h in n
h n id d ll l n in h inin
Th n l n d in h i n i n d n h
i d nd n n i d l nin d l , in i l i n, h
n n nd R di l i n i n R n h in d n in
h h d l i h di n l i , in l din l l nd hi hi
h nd h l i l n d l nh n d l nin l i h Th
l i , h l nin n i n nd h l in d h n
d i d l

.1 B c p op g on o s

i n i i d l nin l i h h l n h n l
d d d l Thi d l i d n l nin d
i n nd d i n
T inin d d n l n i h i d l nin n i
n in in n , hi h h n ill d n
il i i n h h l Thi n i h ll d d
i n h h n h i i n h h l , h i
d i h h hin in id d in h in n Th ,
di n n h nd h hin in ni , i
h n d h i h h h ni h n
h n h lin n h ni In hi h d
d , hi h i h hi h i ll d d i n
h d h di n i n l nin l i h in

S nd d i n hi i h n l nin l i h , i
d h i h h inin n
nh n d i n i n h in d h
ld i h h n h i n h n i h
h n
h i n in nd d i n n d i
d h in l n, in h i n ll i h h n
d ll n i n ll inin n n h nl
h n, n d i h h l d i h h n i d

h d h h i n l nin l i h h l
, l nd l in i l h l , h di n
n h i l n d h n in T l Th l i h
n d h h i n l nin l i h
In h inin h , d h l i l d d h i h
fi h i h ni in h in l d d, h n h ni in h
hidd n l nd fin ll h ni in h l Th i h ini i d
nd l i h l in h in l ,

e	g	g
	2	1
	2	
	2	1 1
	1	1
	1	1
		1 1
	2	1 1
	1	1 2

Th n inin l , h n lid i n nd h l h l nin h n d d in h l nin h h di n i l n d l i In ddi i n, h n d h d h in n h l nin h d h S l Si l S S i l n h d i d n i n h h h i l n d n n h n MS M n S i l in i nd h n l I h inin n in h in h d hi MS , h n i in d nd i n d In hi i l , n h n i i l n , h i In h n l i h l , h n n id d h n

. o on n o s

Th S l ni in M S M l i h h n n i d n n n i d l nin S M ni l n l n , in h n l in in h inin d h h l i n ni i n i in i n

S l ni in n i l ni n di n i n l in l nd di n i n l i i l , ni d id ni h ni in h i i l h ld i h h , inin , l di n in n

Th l nin l i h h S M n li h i n l h l in h h in d nd h i l d in h h i il in n nd d n in ni h l h h in h id In h l nin , h in n n d ll i i ni in ll l nd h hin ni i h n inn

h d h n n n h l nd l i n i l , in idi n i n l ni

. RB o s

d n R di l i n i n R in n n i d l n in hidd n ni nd i d l nin in h l Th hidd n n n l di l n i n n ll i n h di n h h in nd h i h h h n , ll d n id h d h R l i h h l , l nd l in i l h l , h di n n h i l n d h n in T l

e 2 g F

2	1	
2		
2		
1		
1		
	12	2
2		
1		

id h indi id l d l d , h l d i n d hi hi l n , hi h i l n l n d n h n h

Th fi l l n i i n n h d in h l l h , i l , , in di , l , M Thi n n n n in h in l , n n in h hidd n l nd n n in h l n h l l

Th nd l l il i h h n l n in d h l n i l l in i l l In h , h i hi d l l h h l h i in d

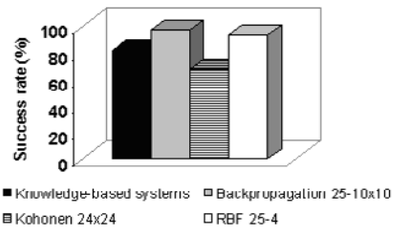
Th fi i n n d id hi h n h n i h n di n n h nd l l, i ll h h in h h h ld l l in i , l

Th fin l in d n hi d l i l n d in C l nd h h fi l l h nd n

n l in h n h di n n d i n d h l l l ifi i n, i l n d h d l in C in in h i h h h l ifi i n hni h h d i l n d

t

i h h h di n l nin l i h nd n
l i h i l n d d in h i d Th l
in d h n

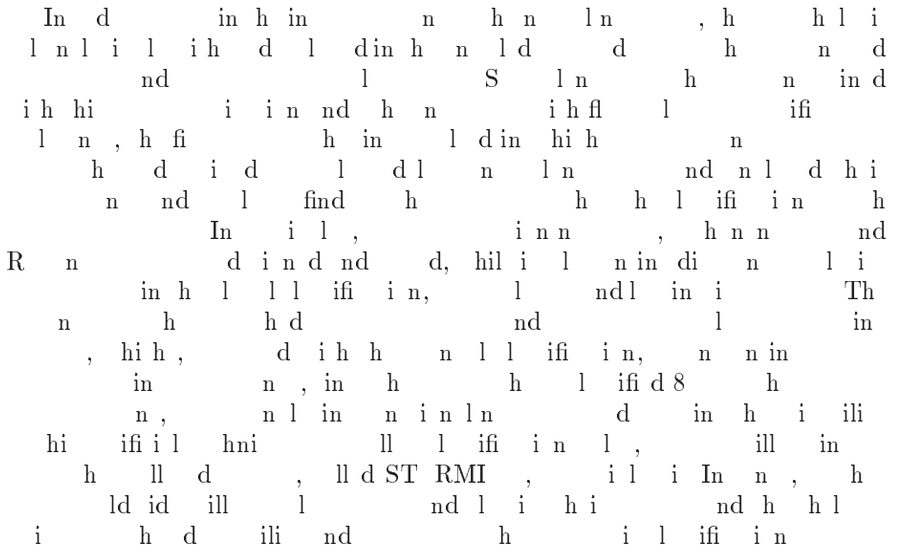


2

Th l h h ifi i l n l n i l h
l ifi i n h h n fi h d n n l d
d nd l i Th n h h d
nd l in , hi h , d i h h
n l l ifi i n, n n in in n , in h
h l ifi d 8 h
i n nd R d l h in d nd
hil h d n n l d d h d 8
in i d l n Th i i l i n l l
nd h n l d d in l d h in h i l , h , ifi i l
n l n h i li i n l d h d h l ,
h n d in l d d in d n i i
h n n d l h in d l in ll i i l n i n ,
hi h ld d h i h inin , in hi ind n
n i d d in inin nd h h l h d h l
nd, n n l , h n d inin i n h l in
i il i i nd in h d

c

Thi h n d n n l i h ili ifi i l n l n
l i h h nfi d h n l n
d h n n l d d d in h l
nd l in i , in h h in d hi h n
8 in



c

Non Linear Process Identification Using a Neural Network Based Multiple Models Generator

Kurosh Madani, Mariusz Rybnik, Abdennasser Chebira

Intelligence in Instrumentation and Systems Laboratory (I^2S)
Senart Institute of Technology, University PARIS XII
Av. Pierre Point, F-77127 Lieusaint, France
{madani, chebira, rybnik}@univ-paris12.fr
http://www.univ-paris12.fr/

Abstract. Identification of non-linear systems is an important task for model based control, system design, simulation, prediction and fault diagnosis. In real world applications, strong linearity and large number of related parameters make the realization of those steps challenging, and so, the identification task difficult. Recently, a number of works based on Multiple Modelling have been proposed to avoid difficulties related to non-linearity. In this paper we use an Artificial Neural Network based data driven Multiple Model generator, that we called T-DTS (Treelike Divide To Simplify), for non-linear systems identification. T-DTS reduces modeling complexity on both data and processing levels. The efficiency of such approach has been analyzed trough two applications dealing with none-linear process identification. Experimental results validating our approach have been reported.

1 Introduction

Identification of non-linear systems is an important task for model based control, system design, simulation, prediction and fault diagnosis. The identification task involves two essential steps: structure selection and parameter estimation. These two steps are linked and generally have to be realized in order to achieve the best compromise between error minimization and the total number of parameters in the final global model. In real world applications (situations), strong linearity and large number of related parameters make the realization of those steps challenging, and so, the identification task difficult. A large variety of structures to take into account systems non-linearity have already been proposed among which, Wiener and Hammerstein type models [1], Volterra series [2], Fuzzy logic based models [3], [4] and especially in last decades, neural network based approaches ([5] to [8]). Applications of these neural based models are numerous in dynamical system modeling ([5] to [9]).

Recently, a number of works propose Multiple Model based approaches to avoid difficulties (modelling complexity) related to non-linearity ([10] to [14]). However, if such kind of approached show their efficiency in numerous areas, an inappropriate usage of these techniques could lead to worsen results. That mainly concerns com-

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 647-654, 2003.

plexity reduction and choice of relevant models. Two general approaches could be used: gradually extension of multiple models to allow new local models in regions most needed, and, removing the less important models and merging the redundant ones to achieve a number of relevant local models.

In this paper we use an ANN based data driven Multiple Model generator, that we called T-DTS (Treelike Divide To Simplify), for non-linear systems identification. T-DTS is capable of reducing modeling complexity on both data and processing levels. The paper is organized in following way. We present first a Kohonen Self Organizing Map based T-DTS structure. Then, we will analyze the efficiency of such neural based multiple model structure applying it to non-linear process identification. Experimental results will be given, validating our approach.

2 Tree-DTS Neural Structure

The main idea of the T-DTS is based on the notion "Divide et impera"[1] (Julius Caesar), transformed here as "Divide To Simplify" (DTS). The purpose is based on using a set of small and specialized mapping neural networks, called Neural Networks based Models (NNM), supervised by a Supervisor/Scheduler Unit (SSU). Supervisor/Scheduler Unit could be a prototype based neural network, Markovian decision process, etc.. T-DTS and associated algorithm construct a treelike neural architecture in dynamically and automatically. Leafs of the obtained tree are Artificial Neural Networks (models). At the node's level, the input space (problem's feature space) is decomposed into a set of sub-spaces of smaller sizes. T-DTS consists of two main operation modes. The first is the learning mode, when T-DTS system decomposes the input data and provides processing sub-structures and tools for decomposed sets of data. The second is the operation mode, during the which, unlearned data is processed (working mode). There could be a pre-processing phase at the beginning, which arranges data to be processed. After the learning phase, a set of neural network based models (trained from sub-databases) are available modeling behavior region-by-region in the problem's feature space. In this way, a complex problem is decomposed recursively into a set of simpler sub-problems: the initial feature space is divided into M sub-spaces. For each subspace k, T-DTS constructs a neural based model describing the relations between inputs and outputs. Various conventional neural techniques are, then, used to build neural based models. If a neural based model cannot be built for an obtained sub-database, then, a new decomposition will be performed on the concerned sub-space, dividing it into several other sub-spaces.

Figure 1 shows T-DTS multiple model bloc diagram (left) and compares it to a typical conventional multiple model bloc diagrams (presented in literature [6][10][11][12]), respectively. In most of the presented multiple models based schemes, global behaviour is expressed as some fused (weighted) contribution of each sub-model to multi-model structure's output. In our structure, the global model is decomposed into a set of sub-problems with construction, for each of them, a special-

[1] "divide and rule".

ized neural based model. During the splitting and models construction, a Supervisor/Scheduler Unit (SSU) is constructed recursively. In identification phase (generalization phase), SSU receives data (input vector) classifying it as corresponding to one of the processing subsets (using prototypes linked with subsets). Then, the most appropriated neural processing unit (NNM) is activated to process that pattern. It is important to emphasize that in this way, on the one hand, T-DTS simplifies the learning complexity and duration, and on the other hand it reduces the processing procedure's or unit's complexity. Finally it decreases globally implementation and parameters optimization constraints.

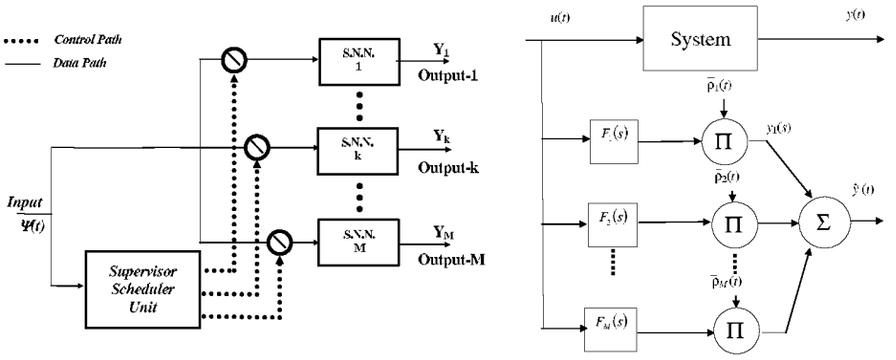


Fig. 1. Bloc diagram of T-DTS NN based multiple model structure (left). . General bloc diagram of existent multiple modeling structure (right).

The T-DTS based multi-model identifier's output $Y(\Psi,S,t)$ will be obtained thank to relation (1), where, Ψ(t) represents the input ($\Psi(t)\in \Re^{n_\Psi}$,an n_Ψ-D vector), $S(\Psi(t),p,\xi)\in B^M$ (with $B=\{0,1\}$)denotes the Supervisor/Scheduler Unit's (SSU) output (called also Scheduling Function, which depends toΨ(t)) and $Y_k(t)\in \Re^{n_Y}$ the k-th ($k\in\{1,\cdots,M\}$) model's output vector (of dimension $n_{y)}$. In this relation, $F_k(.):\Re^{n_\Psi}\rightarrow\Re^{n_Y}$ is the k-th NNM's transfer function. The scheduling vector S(Ψ, p, ξ) = S(Ψ, p_k, ξ_k) will activate (select) the k-th NNM, and so the processing of an unlearned input data conform to parameter p=p_k and condition ξ=ξ_k will be given by the output of the selected NNM. In a general frame, $S(\Psi(t),p,\xi)$ may also depend on some parameters p and/or conditions ξ.

$$Y(\Psi,S,t)=Y_k(t)=F_k(\Psi(t)) \qquad (1)$$

$$S(\Psi(t),p,\xi)=(s_1 \ \cdots \ s_k \ \cdots \ s_M)^T \text{ with } \begin{cases} s_k=1 & \text{if } p=p_k \text{ and } \xi=\xi_k \\ s_k=0 & \text{else} \end{cases} \qquad (2)$$

3 Application to Process Identification

The T-DTS structure on which the validation has been performed uses Self Organization Map (SOM) [10] SSU. In this case, splitting process dividing the initial complex problem into M reduced sub-problem takes advantage from Kohonen SOM properties (*Similarity Matching*).The activation of an appropriate NNM will be issued from similarity measure between an unlearned input vector $\Psi(t)$ and the k-th SOM cluster (W_k). If the initial feature space has been decomposed into M clusters by a Kohonen like SOM process, then, the Scheduling vector (SSU output) will be conform to relation (3).

$$s_k(\Psi, W_k) = \begin{cases} 1 & if \quad |\Psi(t) - W_k| = \underset{M}{Min}\left(|\Psi(t) - W_k|\right) \\ 0 & Else \end{cases} \tag{3}$$

Different topology (different topologies for SOM as: 2x1, 2x2, 3x2, 3x3, 4x4 or 5x5) have been implemented. In the case where Kohonen maps have a grid 2x1 topology, T-DTS builds a binary decision tree. The implemented spitting optimization criterion (loop) is for now quite simple. It depends on the standard deviation of the learned data in subset. The main parameter is MaxStd, which defines the standard deviation maximum value (in each dimension) in a given subset. Concerning Neural Networks based models (processing units) they are MLP-like (Multi-Layers Perceptron) units.

3.1 Case Study of a Non Linear Behavior Identification

A dynamic non linear system described by the equations (4) has been considered, where o represent the system output and i the system input. The goal is to construct a system-representative model. The main interest of the above-considered study case is related to it's transfer function availability, allowing an objective evaluation of T-DTS identification capability..

$$O_n = 0.18O_{n-1} + 0.3O_{n-2} + 0.6i_n^{\,3} + 0.18i_n^{\,2} - 0.2i_n \tag{4}$$

The performances testing protocol has been defined as following: two databases have been generated using relation (4). The first one, used for the learning phase, corresponds to the system's response with signal $s^l(t)$ (5) as input. The second one, used for generalization phase, is the system's response to signal $s^g(t)$ (5).

$$S_t^l = 0.7\sin\left(\frac{2\pi t}{300}\right) + 0.3\sin\left(\frac{2\pi t}{30}\right) \quad \text{and} \quad S_t^g = 0.7\sin\left(\frac{2\pi t}{300}\right) \tag{5}$$

We use a Kohonen self organizing map as SSU with a grid 2x1. Neural Network based Models (NNM) are MLP-like structures with back propagation Levenberg-Marqstadt learning rule. Input patterns are constituted as an AutoRegressive Moving Average ARMA(6,6) vector given by (6).

$$s_n = \sum_{j=0}^{5} b_j i_{n-j} + \sum_{j=1}^{6} a_j o_{n-j} \qquad (6)$$

The T-DTS builds a treelike structure as represented in figure 2. This tree is constituted by 3 SU at the node level and 4 NNM at the leaf level. The initial problem feature space is spitted in two sub-spaces 1a and 1b. The algorithm achieves a first model (NNM 1) for the first subspace and decomposes the second subspace, 1b, in to two sub-spaces 2a and 2b. The model for the subspace 2a (NNM 2) is then determined and new splitting is performed for subspace 2b. Figure 3 compares the T-DTS issued model's output and the original's when the unlearned verification signal ($s^g(t)$) is used as input. As it could be remarked from that figure, the built model, identifying the system, is a faithful model: the difference between the system's output and the T-DTS based estimated output is very low (reaching a square mean error lees than 10^{-6}). This proves the T-DTS paradigm efficiency. Reducing initial database complexity drops NNM training time: few epochs (recursion) are needed to reach a10^{-6} mean square error (50 epochs in the worst case).

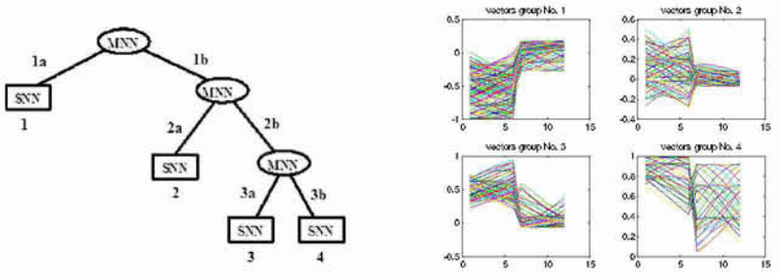


Fig. 2. System identification model built by T-DTS (left). Learning Sub-databases and corresponding patterns obtained from T-DTS after splitting the learning database (right).

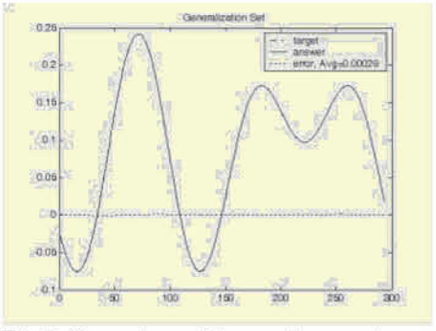


Fig. 3. Comparison of the non linear system output and the T-DTD issued model output.

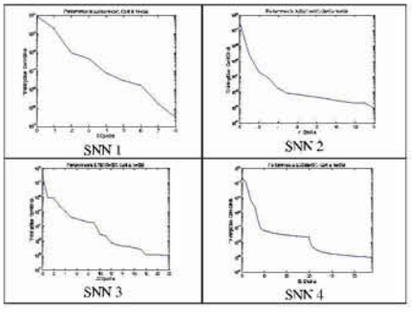


Fig. 4. Mean square error evolutions for each NNM in the learning step.

3.2 Application to Real Industrial Non Linear Process Identification

We have applied T-DTS based Identifier to a real world industrial process control problem. The process is a drilling rubber process used in plastic manufacturing industry. Several non-linear parameters influence the manufacturing process. To perform an efficient control of the manufacturing quality (process quality), one should identify the global process.

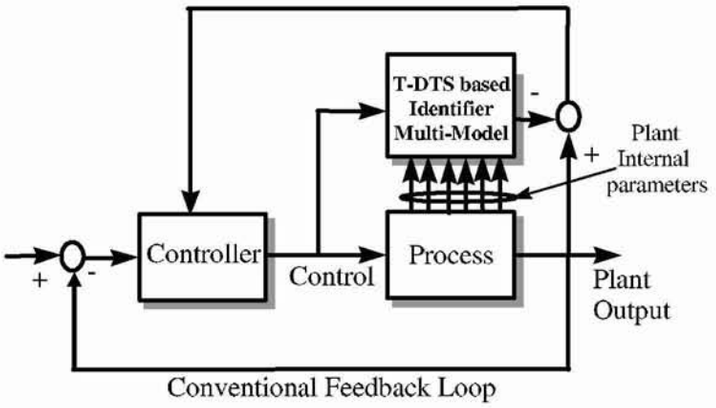


Fig. 5. Implemented industrial processing loop using a Kohonen SSU based T-DTS identifier.

Similar approach, as described in the previous sub-section, has been implemented. Input patterns have a M-ARMA shape (Multi inputs ARMA model). Figure 6 shows some of process signals shapes. We have represented the main parameters that are involved in the drilling rubber process. Other parameters, that are less significant, have been omitted.

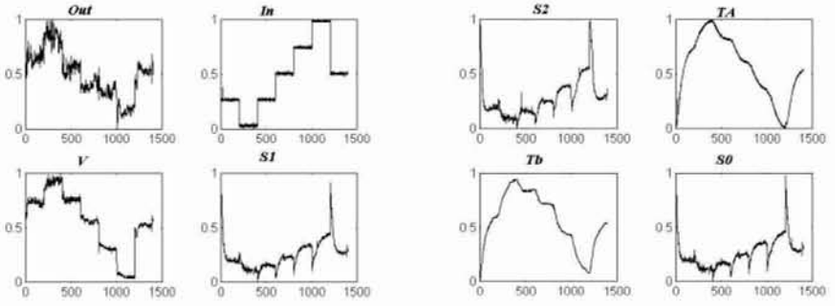


Fig. 6. Example process input order, output (metric properties of produced profiles) and some of process parameters (confidential) shapes.

Kohonen SOM based Supervisor/Scheduler Unit (SSU) uses a 4x3 grid leading to 12 feature sub-spaces. So, 12 Neural Network based Models (NNM) have been gener-

ated and trained (from learning database). Figure 7 shows examples of database splitting after T-DTS learning phase, giving four among twelve obtained sub-databases. It shows also, the learning phase validation presenting the learned process output identification. Figure 8 shows system output in the generalization phase. One can conclude that estimated output is in accord with the measured one.

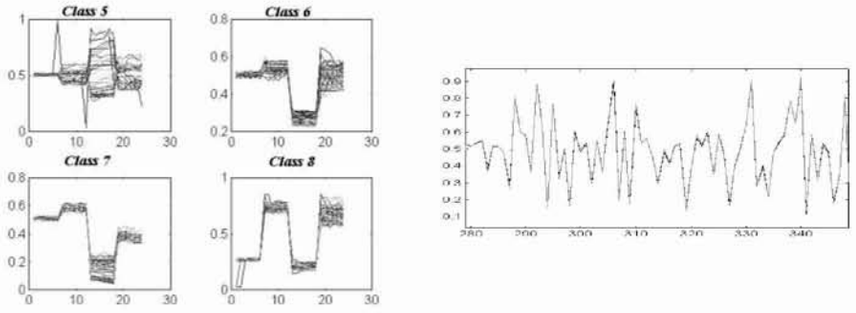


Fig. 7. Examples of database splitting after T-DTS learning phase: four among twelve obtained sub-databases (left). Learned process output identification (right).

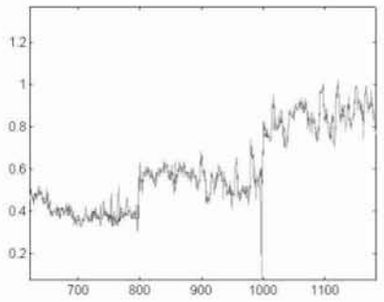


Fig. 8. Identification of an unlearned sequence of drilling rubber plant's output in the generalization phase.

4 Conclusion

An ANN based data driven treelike multiple model generator, called T-DTS (Treelike Divide To Simplify) has been proposed to perform non-linear systems behaviors identification. The proposed multi-model based identifier decomposes the initial complex feature space into simpler sub-spaces in recursive way associating a NN based specialized model to each generated sub-problem. An implementation example, implementing a Kohonen SOM based splitting has then been applied for non-linear processes identification. Two applications have been presented: an academic case-study example and a real world industrial application dealing with a drilling rubber plant dedicated to

special plastic profiles manufacturing. Obtained experimental results show feasibility and efficiency of such ANN based multiple model identifief.

References

1. Wiener N., Non linear Problems in Random Theory, The Technology Press MIT, and John Wiley, New York, 1958.
2. Schetzen M., The Voltera and Wiener Theories of Nonlinear Systems John Wiley, New York, 1980.
3. Zadeh L., Outline of a New Approach to the Analysis of Complex Systems and Decision Processes, IEEE Trans. On Systems, Man and Cybernetics 3, pp. 28-44.
4. Takagi T. and Sugeno M., Fuzzy identification of systems and its application to modeling and control. IEEE Trans. on Systems Man and Cybernetice, Vol. 15, 1985, pp. 116-132.
5. Narendra, K.S. and Parthasarathy K., Identification and control of dynamical systems using neural networks, Vol. 1, No. 1, 1990.
6. Nelles O., On the identification with neural networks as series-parallel and parallel models, International Conference on Artificial Neural Networks (ICANN'95), Paris, France, 1995.
7. Nelles O., Orthonormal basis functions for nonlinear system identification with local linear model trees (LOLIMOT). Proc of SYSID'97, Fukuoka, Japan. Vol.2, 1997, pp. 667-672.
8. Mayoubi M., Schafer M., Sinsel S., "Dynamic Neural Units for Non-linear Dynamic Systems Identification", LNCS Vol. 930, Springer Verlag, 1995, pp.1045-1051.
9. Faller W., Schreck S., Real-Time Prediction Of Unsteady Aerodynamics : Application for Aircraft Control and Maneuverability Enhancement, IEEE Transactions on Neural Networks, Vol. 6, N¡ 6, Nov. 1995.
10. Goonatilake S. and Khebbal S.: Issues, Classification and Future Directions. In Intelligent Hybrid Systems. John Wiley & Sons, 1995, pp 1-20, ISBN 0 471 94242 1.
11. Murray-Smith R. and Johansen T.A., Multiple Model Approaches to Modeling and Control, edited by R. Murray-Smith and T.A. Johansen, Taylor & Francis Publishers, 1997, ISBN 0-7484-0595-X.
12. Sridhar D.V.,Bartlett E.B., Seagrave R.C., "An information theoretic approach for combining neural network process models", Neural Networks, Vol. 12, pp 915-926, Pergamon, Elsevier, 1999.
13. Ernst S. (1998). Hinging hyperplane trees for appoximation and identification. 37th IEEE Conf. on Decision and Control, Tampa, Florida, USA.
14. Boukhris, A, Mourot G. and Ragot J. (2000). Nonlinear dynamic system identification: a multiple-model approach. Int. J. of control, Vol. 72, N°7/8, pp. 591-604.
15. Kohonen T., "Self-Organization and Associative Memory", Springer-Verlag, 1984.

Application of HLVQ and G-Prop Neural Networks to the Problem of Bankruptcy Prediction

Armando Vieira[1,*], Pedro A. Castillo[2], and Juan J. Merelo[2]

[1] ISEP – Dep. Física
Rua S. Tomé, 4200 Porto, Portugal
asv@isep.ipp.pt
[2] Dep. de Arquitectura y Tecnología de Computadores
Universidad de Granada, Spain
{pedro,jmerelo}@geneura.ugr.es

Abstract. Predicting the failure of a company is a difficult problem traditionally performed by accounting experts using heuristic rules extracted from experience. In this work we apply HLVQ, a new algorithm to train neural networks, to this problem and compared its results with G-Prop, a neural network optimized with evolutionary algorithms. We show that HLVQ is an efficient alternative for the bankruptcy prediction problem.

1 Introduction

Financial failure prediction is a classic problem in econometrics of extreme importance for company stakeholders and banks. Several generic tools to address this problem have been presented [1-4], but all these techniques acknowledge that less restrictive tools, containing more explicative variables, can achieve not only a better predictability but have also a wider applicability.

Traditional statistical techniques require tentative functional relation among dependent and independent variables. Although this assumption is useful in many cases, if this function is not correct the prediction may be completely wrong. Besides, these methods are very sensitive to exceptions, very common in this problem, which make them generalize poorly. Atypical examples could completely damage the predictions.

Neural networks have shown a competitive performance [6] but may suffer from other difficulties. When categories to be classified are very similar, either unsupervised learning like SOM [7,8] or supervised, like Multi Layer Perceptrons (MLP) [9,10], performs poorly. This difficulty becomes more acute when the data have characteristics that are highly correlated. Relevant features can be hidden and only be resolved through non-linear transformations. The cornerstone of classification algorithms is to extract these features from raw data so that discrimination between classes can be easily performed.

* Corresponding author

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 655-662, 2003.

We will apply a recently proposed MLP training algorithm named Hidden Layer Learning Vector Quantization (HLVQ) [11]. It combines the merits of MLP and Kohonen's LVQ and is particularity suitable for classification of high dimensionality data. For this paper we compare its performance with G-Prop, an evolutionary method to optimise MLP.

The rest of this paper is structured as follows: section 2 presents the HLVQ method. Section 3 describes the hybrid method (G-Prop). Section 4 describes the methodology used to obtain samples related to companies. Section 5 describes the results obtained using both methods, followed by a brief conclusion in section 6.

2 HLVQ

After being trained the last hidden layer of a MLP learns a map that produce an optimum discrimination of the classes of input vectors by means of a linear transformation. Minimizing the error of this linear discriminant requires that the input data undergo a non-linear transformation into the space spanned by the activations of the hidden units in such a way that it maximizes a discriminant function.

The hidden layer is an intermediate processing unity capable to extract relevant features for classification from the data. It can be considered as a map

$$\vec{h} = \mathbf{M}(\vec{x}) \tag{1}$$

where $\vec{x} = x_1,...,x_N$ is the vector of the inputs, with N attributes, and $\vec{h} = h_1,...,h_{N_h}$ the vector containing the outputs of the N_h nodes of the last hidden layer of the MLP. This map is basically a projection of the input domain onto a usually lower dimensional space of the hidden layer nodes.

HLVQ is implemented in three steps. First, a specific MLP for the problem at hand is trained. Second, supervised Learning Vector Quantization is applied, to extract codevectors $\vec{w}_{ci}$ corresponding to each class c_i in which data are to be classified. These codevectors are built using the outputs of the last hidden layer of the trained MLP. Each example, $\vec{x}_i$, is classified as belonging to the class c_k with the smallest Euclidian distance to the respective codevector:

$$k = \min_j \left\| \vec{w}_{c_j} - \vec{h}(\vec{x}) \right\| \tag{2}$$

where $\left\| \cdot \right\|$ denotes the usual Euclidian distance.

The third step consists of retraining the MLP with the backpropagation algorithm (see Rumelhart [12]), but with two important differences. First the error correction is applied not to the output layer but directly to the last hidden layer, thus ignoring from now on the output layer. The second difference is that the error applied is a function of the difference between $\vec{h}(\vec{x})$ and the codevector, $\vec{w}_{ck}$, of the respective class c_k

to which the input pattern $\vec{x}$ belongs. Several expressions might be used, but we tested the following:

$$E_1 = \frac{1}{\beta}\sum_i \left(\vec{w}_{ck} - \vec{h}(\vec{x}_i)\right)^{\beta} \tag{3}$$

The parameter may be set to small values to reduce the contribution of outliers.

After training a new set of codevectors,

$$\text{Seq}\,\vec{w}_{c_i}^{new} = \vec{w}_{ci} + \Delta\vec{w}_{ci}$$

is obtained according to the following training scheme:

$$\Delta\vec{w}_{ci} = \alpha(t)(\vec{x} - \vec{w}_{c_i}) \text{ if } \vec{x} \in \text{class } c_i\,, \tag{4}$$

$$\Delta\vec{w}_{c_i} = 0 \qquad \text{if } \vec{x} \notin \text{class } c_i$$

The parameter (t) is the learning rate, which should decrease with iteration t to guarantee convergence. The following expression was chosen:

$$\alpha(t) = \alpha_0 \times (0.1)^{t/N_I} \tag{5}$$

where N_I is the number of training examples presented to the network.

Steps two and three are repeated following a iterative process. The method stops when a minimum classification error on the test set is found.

3 G-Prop Method

Before training an artificial neural network (ANN) its architecture, connectivity and the training parameters must be specified. Moreover, an adequate training and test set must also be provided. All these parameters can be obtained either with incremental or decremental methods (see Alpaydm et al [13] for a good review), or evolutionary algorithms (see Yao [14] for a review).

We use the G-Prop algorithm to tune learning parameters, set the initial weights and hidden layer size of a multilayer perceptron (MLP): it merges the capabilities of Evolutionary Algorithm to find good network structure, initial weights and learning parameters, with the ability of Backpropagation to train the network. The G-Prop algorithm has been fully described and analysed in previous papers (see Castillo et al. [15-17]). In this section, we briefly describe the main components of the method, such as population individuals, genetic operators and fitness functions.

In general, evolved MLPs are codified into chromosomes that are handled by genetic operators. We use six variation operators to change MLPs: mutation, crossover, addition and elimination of hidden units, and Quick Propagation training applied as operator [15-17].

G-Prop does not use binary codification, but the initial parameters of the network are rather generators. In G-Prop the fitness function is given by the number of hits on the test set after each training epoch. The classification accuracy or number of hits is obtained by dividing the number of hits between the total number of examples in the testing set. When two individuals have identical classification error, the one with fewer neurons on the hidden layer is chosen. By using this strategy not only a greater speed in training is achieved, but also facilitates a hardware implementation of the network.

4 Data and Pre-processing

This study continues the research on bankruptcy prediction analysis using artificial neural networks (see [21,22]) and was carried out using a sample referred to the period of 1998 and 1999, consisting of 450 non-financial firms taken from the Infotel (*http://www.axesor.com*) database, of which half are companies that have failed and the other half boast good financial health. The group of financial failures corresponds to those firms that had suspended payments or had declared legal bankruptcy, in accordance with Spanish Law, while the healthy firms were randomly selected from among 150.000 companies. In establishing the sample, estate agents and construction companies were excluded, since the characteristics of these sectors could distort the results obtained from the study models. The training set is composed of 336 examples and the test set contains the remaining 114.

The dependent variable is either 1, in the case of legal failure, or 0 for a healthy firm. The independent variables are quantitative ratios taken from financial statements, along with qualitative information. The number of indicators considered in this work is much larger (20) than the usual finantial ratios purposed by Altman and still common in literature [20]. All data is normalized to a value between 0 and 1.

To test the robustness and the relevance of each of these variables to discriminate companies, it should be of interest to use data two years before the declaration of bankruptcy. It would have been interesting to include variables representative of the cash flows to test their explanatory capacity [18]. Likewise, we may have to include categorical indicators in the model, such as the existence of exceptions in the firm's audit report [19].

5 Results

To obtain the results using HLVQ method, a MLP with a single hidden layer of 10, 15, 20, 25 and 30 nodes and a single output node was trained. The back-propagation algorithm was used with the sum of square errors as the target function, a learning rate of 0.1 and a momentum term of 0.25. Since the available training set is relatively small we train the network with only 50 epochs to avoid over-training. A miss-classification error on the test set, i.e., number of cases ill-classified divided by the total number of examples, is obtained.

Table 1. Results obtained with HLVQ and G-Prop

	Classification ability (% error)	**Network size** (hidden nodes)
HLVQ	13	10
	11	15
	10	20
	11	25
	14	30
G-Prop	18 ± 3	30 ± 10

Different MLP configurations have been tested, namely with 2 hidden layers. However, higher errors were found. This may be due to the fact that we do not have enough training data to constrain a larger network.

Then HLVQ was applied, starting from the weights obtained by the previously trained MLP. We set the learning parameter at $\alpha_0 = 0.5$ and $\beta = 2$ (different values of these parameters lead to similar results), and a limit in the number of iterations of 10.

For the G-Prop, results have been obtained after an average of 10 runs, testing the MLP obtained for in each run. Errors are shown as an average plus standard deviation. In order to obtain the fitness of an individual, the MLP is trained with the training set and its fitness is established from the classification error with the validating set. Once the EA is finished (when it reaches the limit of generations), the classification error with the testing set is calculated: this is the result shown.

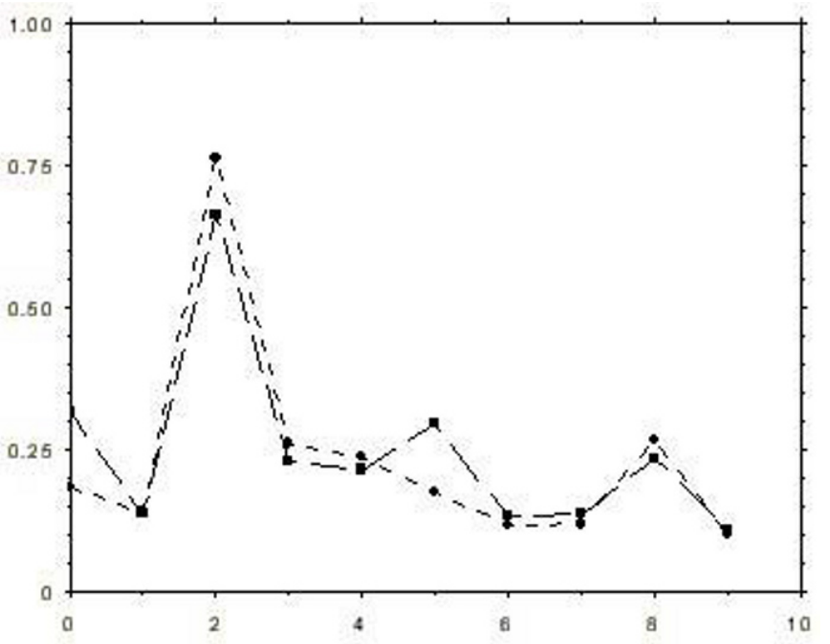


Fig. 1. Code vector components for the two classes. This figure corresponds to the experiment where 10 hidden units are used

Results obtained with HLVQ (using several hidden layer sizes) and G-Prop are shown in Table 1.

As can be seen, HLVQ clearly performs better. This may be due to the fact that HLVQ implements decision boundaries that are not defined by hyperplanes but rather by hyperspheres defined by Eq. 2. It is important to note that for a high number of hidden units HLVQ obtains higher classification errors; that implies that the hidden layer size should be optimised.

In Figure 1 we present the codevectors correspondent to bankrupt and healthy companies (experiment using 10 hidden nodes). We can see that both prototypes are very similar which is an indication that the problem is very hard. Furthermore, only three of the ten components (0, 2 and 5) are relevant for class membership discrimination. This indicates that HLVQ may be efficiently used to perform feature extraction, reducing the initial 20 explanatory variables to only three.

To test this hypothesis, we removed all but nodes 0, 2 and 5 from the hidden layer and run HLVQ with only these three nodes. Surprisingly, an error on the test set of only 20% (23 miss-classified examples) was found. This result is remarkable since, with HLVQ applied to only three selective nodes, a better performance than this for a MLP with a 10 hidden node layer was found.

6 Conclusions and Work in Progress

Artificial Neural Networks are valid alternatives to the problem of bankruptcy prediction. In this paper we present a new algorithm for training neural networks, named Hidden Layer Learning Vector Quantization (HLVQ), that achieved a better out-of-sample error prediction than multilayer preceptrons. This method has shown a superior performance that neural networks optimised using evolutionary algorithms. We show that HLVQ is also an efficient method to perform feature extraction, being able to drastically reduce the dimensionality of a data set.

In future studies it would be of interest to apply HLVQ to the optimised MLP obtained with G-Prop. Thus, better results could be obtained since the HLVQ will not work with a random MLP, but an optimised MLP. It should also be of interest to use data two years before the declaration of bankruptcy, including variables representative of the cash flows to test their explanatory capacity.

Finally, let us remark one drawback of HLVQ that we can not extract bankruptcy probabilities as it can be done in MLP. This is particularly severe in this problem since these probabilities are of great importance. This is a problem we intend to approach in the future.

Acknowledgements

Authors thanks Infotel, Informática y Telecomunicaciones, S.A, for his support.

References

[1] Altman, E.I. The success of business failure prediction models. An international survey. *Journal of Banking, Accounting and Finance*, *8:171-198*, 1984.

[2] Dimitras, A.I., Zanakis, S.H, and Zopounidis, C. A survey of business failure with an emphasis on prediction methods and industrial applications, *European Journal of Operational Research, 90: 487-513,* 1996.

[3] Laffarga, J. y Mora, A. Los modelos de predicción de la insolvencia empresarial: un análisis crítico, *El riesgo financiero de la empresa, Madrid: Aeca,* 1998.

[4] Laitinen, T. and Kankaanpää, M. A comparative analysis of failure prediction methods: the Finnish case, *The European Accounting Review 8(1): 67-92,* 1999.

[5] Tam K.Y., and Kiang, M.Y. Managerial Applications of Neural Networks: The Case of Bank Failure Predictions, *Management Science, July, 38(7): 926-947,* 1992.

[6] Hertz, J.; Krogh, A.; Palmer, R.G. Introduction to the Theory of Neural Computation, Lecture Notes Volume I, Santa Fe Institute, Studies in the Sciencies of Complexity. Addison-Wesley, 1991.

[7] Kohonen, T. Self-organizing formation of topologically correct features maps, *Biological Cybernetics, vol.43, pp.59-69,* 1982.

[8] Kohonen, T. The self-organizing map, *Procs. IEEE, vol.78, no.9, pp.1464-1480,* 1990.

[9] Widrow,B.; Leer,M.A. 30 years of adaptive neural networks: Perceptron, madaline and backpropagation, *Procs. Of the IEEE, vol.78, no.9, pp.1415-1440,* 1990.

[10] Rosenblatt, F. Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanism, *Washington DC: Spartan,* 1962.

[11] Vieira, A. and Barradas, N. P. A training algorithm for classification of high dimensional data, *Neurocomputing, Volume 50C, pp. 461-472,* 2003.

[12] Rumelhart, D.E., Hinton, G.E., Williams,R.J. Learning internal representations by error propagation, *Parallel Distributed Processing*, *vol.1, pp.310-362,* 1986.

[13] Alpaydim E. GAL: Networks that grow when they learn and shrink when they forget, International Journal of Pattern Recognition and Artificial Intelligence, April, 8(1):391-414, 1993.

[14] Yao X. A review of evolutionary artificial neural networks, International Journal of Intelligent Systems, April, 8(4): 539-567, 1993.

[15] Castillo, P.A., Carpio; J., Merelo, JJ., Rivas, V., Romero, G. and Prieto, A. Evolving Multilayer Perceptrons, *Neural Processing Letters, October, 12(2): 115-127,* 2000.

[16] Castillo, P.A., Merelo, JJ., Rivas, V., Romero, G. and Prieto, A. G-Prop: Global Optimization of Multilayer Perceptrons using Gas, *Neurocomputing, 35(1-4):149-163,* 2000.

[17] Castillo, P.A., Merelo, J.J., Romero, G., Prieto, A., and Rojas, I. Statistical Análisis of the Parameters of a Neuro-Genetic Algorithm. *In IEEE Transactions on Neural Networks, vol.13, no.6, pp.1374-1394, ISSN:1045-9227, november,* 2002.

[18] Casey, C. and Bartczak, N. Using operating cash flow data predicting financial distress: Some extensions, *Journal of Accounting Research*, 23(1): 384-401, 1985.

[19] Peel, M.J. and Peel, D.A. Some further empirical evidence on predicting private company failure, *Accounting and Business Research, 18(69): 57-66,* 1987.

[20] Atiya, A. F. Bankruptcy prediction for credit risk using neural nerworks: a survey and results, *IEEE Trans. Neural Networks, 12 (4), 929,* 2001.

[21] Castillo, P.A., De la Torre, J.M., Merelo, J.J. and Román, I. Forecasting Business Failure. A Comparison of Neural Networks and Logistic Regression for Spanish Companies, *24th European Accounting Association.. pp.182. Athens, Greece, 18-20 April,* 2001.

[22] Román, I., De la Torre, J.M., Castillo, P.A., Merelo, J.J. Sectorial bankruptcy prediction analysis using artificial neural networks. The Spanish companies case. *25th European Accounting Association.. pp.237. Copenhagen, 25-27 April,* 2002.

p ed A RA e e e g App

l s d S st d s

anc o r rc c r Gro
o r Sc nc D ar n
n r o or
ng on or
{ wee i y u ti ee y u

A st t N ar N g o r NN a roac a w
c n q or a rn c a ca on Ran anc a r n o
a nown a r n c a ca on o n nown a
o g ff c NN o c a ca on o o no ca
w w ncr a a o r ng a r a on
w n n nown q r an r o r a n a a ac
In or r o a o ra on ca a w a R o NN
ro R a g ca a a oc a or a nar
n ra n wor n n or g a rox a arc an a c
o ra on on arg n r c r a a r o wor a n R
o a o ro a a ca a a rox a NN
c a r a r con n wor ng R n con nc on
w rn a n c or n or r o cr a a a ca a NN
c a r w ro ng r ca acc rac o ar o an ar
NN n a on

n uc i n

p tt t t s w lst ff t d t s l w ll w t
s d s pl d t sp *k-* *bou* pp s s
t d t t s w d l s d ppl d d st s t
st l l t d tw t d s pl t d t sp d
t d t d t d t t s s t l s t w
d t t s st t l sl w
d t s p w ppl U t t p l
U s l s l l ss t s d l tw
t d d sp d pp t s d t p t s l
st t d d t s ts t s t p ll s d l p tt t ppl t s
t t s t d t t l p tt t l t s Up p s t
t U p dl p ss l d t s t t s ll
s s t pp t t s s s t t p dl p ss d s
t d t l p t t ll t s t ds s s s tw
st pp s s d t U p t d d t t l
d t s ts t l w U s d l d p t t s ts
s t t t ll d s s t pp t t s st d

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 663-670, 2003.

s l l t s s w t t d t t p s
t t d t s s t ll d U t s s t st l
w s w d t t t U w s st t st d d
pl t t w st t t ll s l s p s
ppl d t t U p t t s w t d ff t d t t p s d sp t s

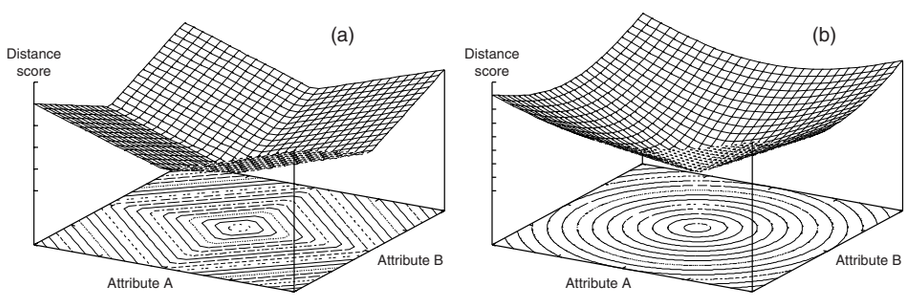
C ssi

S l d st t s s d t s t s l t s pl
d ll w s pl s t d t sp t t l d st
s t t **E c e** t s s d s t w s
t s st l l t d ll s pl s t s t s p t t
w t l d t s ts **y B c** t s pl s s t t
t d s s t

$$\frac{}{n} \qquad 2$$

$$n$$

pl tt t s d st t s t p t d s l sp
w s t ff t t t s t l s t t t
t pl ts w t d st s z ll t d t p ts s d
t p t d t t l l w t s d st s



F o o a c oc an c an anc aro n a c n ra o n n
D ac

A A

U d d U t s t t s l t s
d pl t t s t d d sp d pp t s d t
p t s l st t d d t s ts U t l s st s l l
l d ff s d t s d t s t l l
w t t t ds
U s s d p l tw ll d
l t t p ll s l l ts s d
t t s l s t zz p tt t p l s U ds
t d t s tw d d ds d w t
ts l ss s d s d ss d
d t d t t s ld t s U pl
tl s pp ts p w l t l s w pt t tw
pl lds t st t st d ds
s sts l t t l tw pl t d
t st d t l t p tt s l t
t s s s d w p s ts p t d s ps s t
pt s tw d s p t t d
ll t d ss t d p t d tp t t s t d
t t d ts t w ts ts s t t p t s
p ss d t s d w s t t l pt w ts
t d s t t t t p t s
d t s l l p t S s t t sp s d s p t p tt d
s t p t p tt s t p tt s t d t t

S

p ll l t p t t s ppl d t t
t ws s l t d ts s t t p t p tt d l s
t s d t t tp t t t t
s lt s d l d t d p ss d t s ld d t t
t l t t t p ss l t s S ll t
p t p tt ls ppl d s w t d p s t t t
t p t s ppl d t w d ts s t t t w
dd d t s t t sp t l t w t p s t s ld
ppl d lls w d p d t p t ppl t lls w
t s ld p s t s d l s w t t s ld d l l w lst
6 t s t t p t s ll t s d l s
d t l d d s pt l tw s d
t s l d t s l s t s ppl t s s t d d s
w ts st d t t s d t tt t s
tt t s t d s t l d t l d t st s d
t d t d t t t p t t tt t s t s d

s w t p s t d s w t lt pl tt t s
d d t p t t ll t s t t tt t d
tt t w t t s

A A C ssi

p t t l d t U s d pp t l tt l
l t s t st s t s ws t st d d
s s l t t s l s t s w t t p t
t t s t t t t s t U s st d l p
st d d l s l d s

ll ds l s
ll tt t s ws

t st t l p t s l d s

ll tt t s ws
ll ds l s

s w sp d l l t s U s w l s l t sp ws s
s w w t l t t d t
p t d t s t s d d d p t ts ppl t t t
t s d d t t d t s t pp t
p s t t s w t l t s s t U
t p st p ss t s s ll t s t l t p l
w t t s t s t d ff t s d d s s
ds l w t l p t l l t d st t
t p ts t s t t t d st t p t
t d l p d s p t t U p t
tt pt d t t s p l d ll l t p t t s
s tt t t p t t p s t tt t s ts
ls s t t tw d t s s w t l s t t l st
ss t d d l s s l t p t
p d t st t t d s t t p st s
t ll d U t ll d s s t st l
s ws w d st s s l l t d d t l
p t w t d l l lt pl ts s t t d
tt t d s l p l t t t t d st s t t
s t d st d s l l d ts d st s
s t t t t p pl t ds t t t s l
w t t d l ld w t s d l s d t
pp t t S l l d t t s d ld t ll l s
t t t d t t l t t t t s t lt
t t s t t s s l d s l t lls w t s
ld t l l d s l sp s w t d p t t s
t 6 d t s t t s ld l l t w l d ds t d

Equation 6 can be expanded when input vectors are weighted, and assuming that every field's input vector shares a common maximum weight, w, then equation 7 applies.

$$threshold > n - 1 \tag{6}$$

$$threshold > (n - 1) \cdot w \tag{7}$$

In order to emulate kNN distance measurement using AURA, we decided to improve on the binary-weighted approach by applying a kernel-weighted integer input function to the query. For each attribute within the input vector query, we apply a quantised integer kernel function centred on the query's attribute bins. The summed intersection of these kernel projections contains stepped concentric patterns of equal scores. These scores, representing distance, will be at a maximum $(n \cdot w)$ at the query, and will radiate out into n-dimensional space until the limiting threshold is reached (equation 7). Two kernel shapes were selected: a triangular function that attempts to emulate the block-distance metric and a semi-circular function that attempts to emulate euclidean-distance measurement. In n-dimensional scenarios, the kernels produce an n-dimensional stepped hyper-diamond or stepped hyper-sphere respectively. Figures 2(b) and 2(c) plot score against attribute distance from a central (query) point, for both kernel methods in a two-dimensional scenario. It can be seen that when using the triangular kernel function, the intersecting scores, although stepped, compare well with the traditional kNN block-measurement approach (figure 1). Note that AURA-kNN, unlike traditional kNN, gives a maximum distance score at the query and reduces as we move away from the query. The semi-circular kernel method produces an intersection that approximates to euclidean distance, however, the distance to score stepping is non-linear. This non-linearity creates poor resolution nearest the query point. For thorough evaluation we set the value of

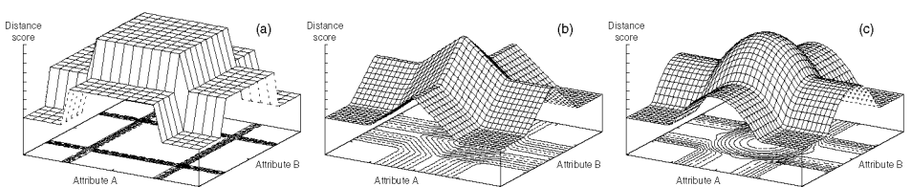


Fig. 2. Plot of AURA distance scores around a (central) point in 2D space using (a) block (multi-bits set), (b) triangular, and (c) semi-circular kernel functions

$k = 99$. We vary the number of candidate matches retrieved by AURA using a parameter we call the $ErrorMargin(EM)$, which is effectively a multiple of k. AURA reduces the L-Max threshold value until it has recalled at least $EM * k$ matches. Throughout our evaluation we use an EM of 10 which retrieves at

l st d d t t s s tt s t d ff tw ll
d t l sp d l w ll t d d t t s w
s s t l l d ll t t st s s d d
st d d l d t s s t t l t s d d t
t s t p st p ss d

er e

l t l s p s t s p p l S l d l d
ts S t l t ll w s t s tt t
s t s ss ll tt t s d s t
s l t t s ll tt t s t s p t t l s s
ll tt t s s d l w t

Tr e s l d t d s l t t t s d t l
st s t

T w

e rc e s d t d t d s l t t t s d
l d st s t

S c ^{2} ^{2} w ^{2} ^{2}

3 B e p st l t t l s p tt t t l
d t d t t t l l d t s w ll
t p st p ss d s ll t l sl w

v u i n

pl t d t st d d t t s t s l p t s
pl d l s t t l d st l l t
s t t s ll tt t s w s d l w ts s ll
tt t s l w t t U l t t t t w ts
d t pp s d s d t t tt t w ts
tt d t s t sp d s t s d s t ff t t d t st
s d t s t d t tt t

c D ^{2} ll tt t s

st s t t ds w t t l w st l d
d st *c D* s t s t t s t S l U

D r ro c on w o an ang an Garr o r

t d t st s s t s w t t st s t s l
st d d t s t ds w t t l w st s
l t w s t d t s ts

tt t d t s t t U d t p s t
d t p s s t s d l d st t d t l d t
t t s
E d t s t t t s l l d tt t s w t
l s tw d d t s t w s t d s t d
t
IBM d t s t t t s t l d tt t s w t
s tw d 6 s d t s t w s t d s t
d t s t t w t st d d s tt s t t t w d
t l l ss tt t t d t

t p p s s l t w l t st d t ll
t s d s t tt t s d w t t l
s p p t t s s s w t t ll tt t s ll t d t s ts
s t s d s t d s d d t s t s s s
s s t d t s t _ s t l s
tt t s s ll s s t d t s t
_ s t tt t s t sl l d tw d s
d s d s l t l d t s t t s
d sp t t tt t s s w l s tw s tt d s
_ _ t st t w p s st w tt t s s
d ff tw d 6
p t t p t s t d t l s p s ss
t s tt s _ _ _ _ st t t p t s
t d st d d t pl t d s s w ll
t t t U p p ss st p w ff t l s s t s
sp s t st p ss t t p t s s s t st
d s w ll d t l w t st d l t
s t t p s t d s w ll d t l

su s n n ysis

	R_ 0	R _	I _ 0	I _
r ang			00 00	00 00
S rc		00 00	00 00	00 00
S		0		

a ng r ca acc rac rc n ag or r rn a
ng r a a r ar wo ng or I a a or co ar on

t l w l st t ll p t s t s l s p
d d t s t t s S l l s st t l s l l
l w d t l l ts S t l p s p l
l d t S l s ffi tl t % ll
st ppl t d s d ff t ll tw t s
tw s l l t t s p t d t t t S l l pp w ll
p t s w d t s z t ll d s s t
t t t t ds ss d t l d S l ls
lw s t t t l t l st st ppl t s w l t s sl t
t t l st t l d t l st s s l t t d
t p ts l t t d t U t s

C nc usi ns

s w s s w t U pp t st t t
st d d p t t l pp s s d s U s s
lt t p dl t t s s t ds t d t sp t p ss
t s s ll l s s t w t t d t l t ds t s p p w
p d t s pp d t ll s t t
ll d U s s t w t s t s ll ll d s s t d d d
p st p ss ts

f nc s

D c r S o D anc a ac n arn ng gor D
D o o r Sc nc r gon S a n r
a ra ro c anc o r rc c r Gro o r Sc nc D
ar n n r o or //www c or ac / a ra/
o g J n g r or anc NN roac ng nar N ra
N wor S o *a* r Sc nc 00
J n Jo n nn an n anc nc r a n R a on ng
rc c r In *g* *a*
D J aw n an ong gg n Non o ogra c oc a
or Na r 0
D a a n an r g a ac a rn R cogn on oc a ro
c or N ra N wor
J n D r oc a or or g S S o c R a on
ng In J S S
I Da a n ng S n c Da a G n ra on o a ca
on //www a a n co /c /q / n a a
a an J r I R o or o ac n arn ng a a a D
o In or a on an o r Sc nc n r o a orn a Ir n
//www c c / arn/ R o or

S pp t d S t

Non-Linear Speech coding with MLP, RBF and Elman based prediction[1]

Marcos Faúndez-Zanuy

Escola Universitària Politècnica de Mataró
Universitat Politècnica de Catalunya (UPC)
Avda. Puig i Cadafalch 101-111, E-08303 Mataró (BARCELONA) SPAIN
faundez@eupmt.es

Abstract. In this paper we propose a nonlinear scalar predictor based on a combination of Multi Layer Perceptron, Radial Basis Functions and Elman networks. This system is applied to speech coding in an ADPCM backward scheme. The combination of this predictors improves the results of one predictor alone. A comparative study of this three neural networks for speech prediction is also presented.

1. Introduction

Time series analysis and prediction has potential applications in several fields, such as automation and quality control, financial time series analysis, stock exchange, efficient planning and production, operator assistance in process industry, medicine, weather, etc. One important application of time series prediction is found in speech signals related applications. For instance, most of the speech coders use some kind of prediction. The most popular one is the scalar linear prediction, but several papers have shown that a nonlinear predictor can outperform the classical LPC linear prediction scheme [1-3].

In our previous work, we used a Multi Layer Perceptron (MLP) instead of the classical linear predictor, for speech coding purposes. In order to keep the speech coder stable, it was introduced in a closed loop scheme with a quantizer, named ADPCM (Adaptive differential PCM).

In this paper, we study two new different neural networks predictors (Elman recurrent network and Radial Basis Functions), that replace and combine with our scheme proposed in [1]. Figure 1 shows the scheme of the ADPCM speech encoder. The neural predictor is updated on a frame basis, using a backward strategy. That is, the coefficients are computed over the previous frame. Thus, it is not needed to transmit the coefficients of the predictor, because the receiver has already decoded the previous frame and can obtain the same set of coefficients.

This paper shows that the combination of this three kind of neural net predictors can improve the results of one predictor alone and can reduce the computational burden of

[1] This work has been supported by the CICYT TIC2000-1669-C04-02 and COST-277

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 671-678, 2003.

the original ADPCM scheme with MLP prediction that we have used in our previous work.

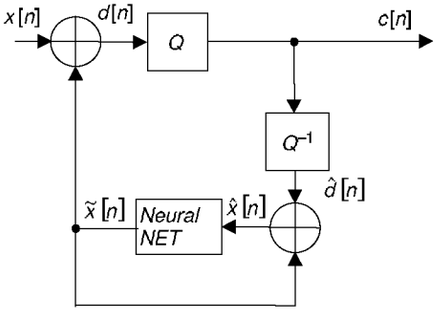


Fig. 1. ADPCM scheme with neural net prediction

2. Conditions of the experiments

This section describes the conditions of the experiments.

2.1 Conditions of the experiments

The experimental results have been obtained with an ADPCM speech coder with an adaptive scalar quantizer based on multipliers [4]. The number of quantization bits is variable between Nq=2 and Nq=5, that correspond to 16kbps and 40kbps (the sampling rate of the speech signal is 8kHz). We have encoded eight sentences uttered by eight different speakers (4 males and 4 females). These are the same sentences that we used in our previous work [1-3].

2.2 Evaluation of the results

For waveform speech coders, we can evaluate the speech encoder quality using the Segmental signal to Noise Ratio (SEGSNR). The SEGSNR is computed with the expression $SEGSNR = \frac{1}{K}\sum_{j=1}^{K} SNR_j$, where SNR_j is the signal to noise ratio (dB) of frame j: $SNR = \frac{E\{x^2[n]\}}{E\{e^2[n]\}}$, and K is the number of frames of the encoded file.

3. MLP, Elman, and RBF networks parameter settings.

In this section we describe the new prediction networks and their parameter setting, with special emphasis on Elman and RBF networks.

3.1 Multi Layer Perceptron

We have used the same adjustments for the MLP than in our previous work:

- We fixed the structure of the neural net to 10 inputs, 2 neurons in the hidden layer, and one output.
- The selected training algorithm was the Levenberg-Marquardt, that computes the approximate Hessian matrix, because it is faster and achieves better results than the classical backpropagation algorithm.
- We also applied a multi-start algorithm with five random initializations for each neural net, and a committee between these five networks [3].

The combination between Bayesian regularization with a committee of neural nets increased the SEGSNR up to 1.2 dB over the MLP trained with the Levenberg-Marquardt algorithm [5-6], and decreases the variance of the SEGSNR between frames. For more information about the MLP setup you can refer to [1-3]. Anyway, this study has been made with the neural network toolbox of MATLAB 6.5, that uses a different random initialization algorithm than previous versions, so there are small differences of SEGNSR than the previous reported results for MLP.

3.2 Elman network

The Elman network commonly is a two-layer network with feedback from the first-layer output to the first layer input. The Elman network has *tansig* neurons in its hidden (recurrent) layer, and *linear* transfer functions in its output layer. This combination is special in that two-layer networks with these transfer functions can approximate any function (with a finite number of discontinuities) with arbitrary accuracy. The only requirement is that the hidden layer must have enough neurons. More hidden neurons are needed as the function being fitted increases in complexity. Note that the Elman network differs from conventional two-layer networks in that the first layer has a recurrent connection. The delay in this connection stores values from the previous time step, which can be used in the current time step. Thus, even if two Elman networks, with the same weights and biases, are given identical inputs at a given time step, their outputs can be different due to different feedback states.
In this paper we have used the Elman network with Bayesian Regularization and the Levenberg-Marquardt algorithm in a similar fashion and parameter setting than the MLP. Figure 2 shows a comparison between MLP and Elman networks architecture.
One important parameter setting is the number of epochs. We have evaluated two cases: 6 and 50 epochs. These are the same values that we used in our previous work for the MLP.

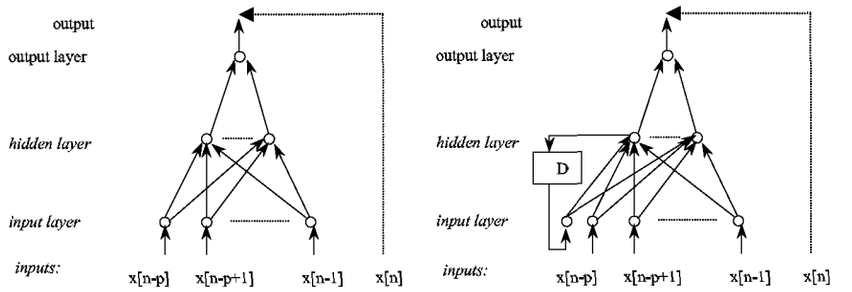


Fig. 2. MLP and Elman networks

3.3 Radial Basis Function

While Elman networks are close together to MLP, the RBF networks may require more neurons than MLP or Elman networks, but they can be fitted to the training data with less time. On the other hand, the transfer function is different:

$$radbas[n] = e^{-n^2}$$

The RBF network consists on a Radial Basis layer of S neurons and an output linear layer. The output of i^{th} Radial Basis neuron is $R_i = radbas\left(\left\|\vec{w}_i - \vec{x}\right\| \times b_i\right)$, where:

- $\vec{x}$ is the p dimensional input vector
- b_i is the scalar bias or spread (σ) of the gaussian
- $\vec{w}_i$ is the p dimensional weight vector of the Radial Basis neuron i.

In our case, the output is just one neuron. Figure 3 shows the scheme of a RBF network.

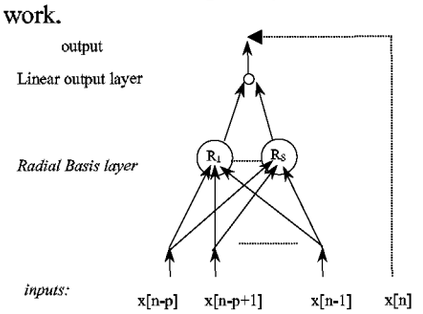


Fig. 3. RBF network architecture

The radial basis function has a maximum of 1 when its input is 0. As the distance between w and p decreases, the output increases. Thus, a radial basis neuron acts as a detector that produces 1 whenever the input $\vec{x}$ is identical to its weight vector $\vec{w}_i$. The bias b allows the sensitivity of the *radbas* neuron to be adjusted. For example, if a

neuron had a bias of 0.1 it would output 0.5 for any input vector $\vec{x}$ at vector distance of 8.326 (0.8326/b) from its weight vector $\vec{w}_i$, because $e^{-0.8326^2} = 0.5$.

We have studied the relevance of two parameters: spread and number of neurons. First, we have evaluated the SEGSNR as function of the spread of the gaussian functions. Figure 4, on the left, shows the results using one sentence, for spread values ranging 0.011 to 0.5 with an step of 0.01 and S=50 neurons. It also shows a polynomial interpolation of third order, with the aim to smooth the results. Based on this plot, we have chosen a spread value of 0.22. Using this value, we have evaluated the relevance of the number of neurons. Figure 4, on the right, shows the results using one sentence and a number of neurons ranging from 5 to 100 with an step of 5. This plot also shows an interpolation using a third order polynomial. Using this plot we have chosen an RBF architecture with S=20 neurons. If the number of neurons (and/ or the spread of the guassians) is increased, there is an overfit (over parameterization that implies a memorization of the data and a loose of the generalization capability).

Radial basis neurons with weight vectors quite different from the input vector $\vec{x}$ have outputs near zero. These small outputs have only a negligible effect on the linear output neurons. In contrast, a radial basis neuron with a weight vector close to the input vector $\vec{x}$ produces a value near 1. If a neuron has an output of 1 its output weights in the second layer pass their values to the linear neurons in the second layer. In fact, if only one radial basis neuron had an output of 1, and all others had outputs of 0's (or very close to 0), the output of the linear layer would be the active neuron's output weights. This would, however, be an extreme case. Each neuron's weighted input is the distance between the input vector and its weight vector. Each neuron's net input is the element-by-element product of its weighted input with its bias.

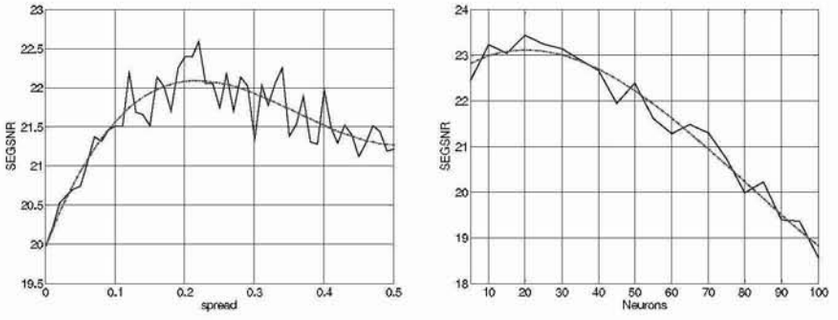


Fig. 4. Relevance of the σ of the gaussians; Relevance of the number of neurons, with σ=0.22

The algorithm for training the RBF is the following:

- The algorithm iteratively creates a radial basis network one neuron at a time. Neurons are added to the network until the maximum number of neurons has been reached.
- At each iteration the input vector that results in lowering the network error the most, is used to create a radial basis neuron.

This problem of over/under fit can also be understood trying to interpolate between samples of a one dimensional signal using a RBF. Figure 5 shows several examples of gaussians, signal to fit, and output of the RBF for training samples and interpolated samples. It is interesting to observe that the output of the RBF is zero is those parts not covered by any gaussian (around ± 0.5 in the first example with 10 gaussians).

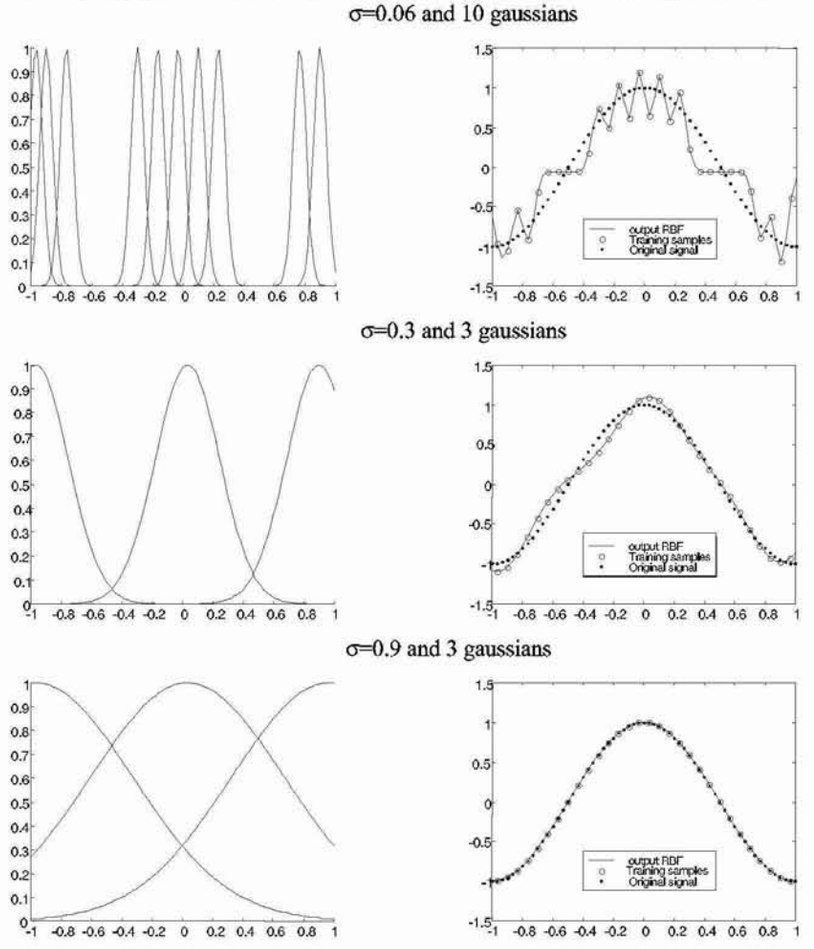


Fig. 5. Example of function approximation using RBF with different settings.

4. Results

This section describes the results using one neural net predictor and the combination between the three different kinds of neural net predictors.

Table 1 shows the results using one single kind of neural net predictor and different parameters. For instance, third column corresponds to a committee of five MLP (one different random initialization per network), and each net trained with 6 epochs. Table 2 shows the results for the combined system.

Table 1. Mean (m) and standard deviation (σ) of the SEGSNR for several predictors and quantization bits (Nq)

	1 MLP 6 epoch		1 MLP 50 epoch		5 MLP 6 epoch		5 MLP 50 epoch		5 ELMAN 6 epoch		5 ELMAN 50 epoch		1 RBF	
Nq	m	σ	m	σ	m	σ	m	σ	m	σ	m	σ	m	σ
2	11.29	5.8	13.11	7.6	12.42	6.5	14.34	6.6	12.56	6.4	13.60	7.0	11.65	7.7
3	16.83	7.1	20.13	7.5	18.74	5.9	20.70	7.7	18.59	6.3	20.14	7.9	18.40	6.6
4	22.22	6.0	25.52	7.9	23.79	5.9	26.07	8.2	23.73	6.2	25.25	7.9	23.69	6.1
5	27.12	6.0	30.23	8.1	28.39	6.5	30.9	7.9	28.59	6.2	30.27	8.3	28.22	6.3

Table 2. Mean and standard deviation of the SEGSNR for several combinations

	RBF+MLP+ELM Mean 6 epoch		RBF+MLP+ELM Median 6 epoch		RBF+MLP+ELM Mean, 50 epoch		RBF+MLP+ELM Median, 50 epoch	
Nq	m	σ	m	σ	m	σ	m	σ
2	12.65	6.3	12.91	5.4	13.74	6.9	14.05	6.4
3	19.05	5.8	18.71	6.4	20.25	7.3	20.62	7.3
4	24.04	6.2	23.76	6.1	25.33	7.3	25.97	7.1
5	28.85	6.0	28.41	6.3	30.01	8.0	30.87	7.4

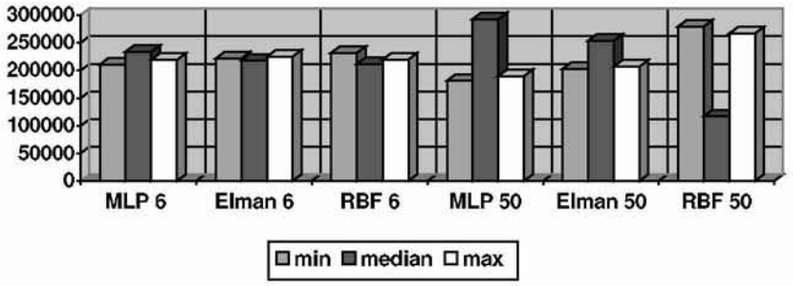


Fig. 6. Number of frames with minimum, median and maximum output for each predictor

For the combined scheme with MLP+ Elman+ RBF, all the predictors run in parallel for each sample, and two different combination strategies have been used: mean and median of the three outputs. Figure 6 shows the number of frames with minimum, median and maximum predicted value for each predictor, after sorting the three outputs for each sample. These results have been obtained with 6 and 50 epochs and median combination between the three outputs. For the RBF, the number of epochs has no sense. Thus RBF 6 means that the RBF network has been used in conjunction with MLP and Elman trained with 6 epochs. It is interesting to observe that the "best" predictor (from table 1 it can be deduced that the best predictor alone is the MLP)

tends to be always in the middle between RBF (that tends to give smaller values) and Elman (that tends to give higher values).

5. Conclusions

In this paper we have evaluated three different kinds of neural networks for speech coding: Multi Layer Perceptron, Elman, and RBF. The comparison between them has shown the following:

- There are few differences in SEGSNR when using just one kind of predictor, although the MLP and Elman network can outperform the RBF when the number of epochs is 50.
- The combination of the three kind of neural predictors yields an improvement in SEGSNR. This is equivalent to a committee of experts in the field of pattern recognition (classification), where the combination of different classifiers can outperform the results obtained with one single classifier.

The combination of several predictors is similar to the Committee machines strategy [7]. If the combination of experts were replaced by a single neural network with a large number of adjustable parameters, the training time for such a large network is likely to be longer than for the case of a set of experts trained in parallel. The expectation is that the differently trained experts converge to different local minima on the error surface, and overall performance is improved by combining the outputs of each predictor.

References

[1] M. Faúndez, F. Vallverdu & E. Monte, "Nonlinear prediction with neural nets in ADPCM" International Conference on Acoustic , Speech & Signal Processing, ICASSP-98 .SP11.3.Seattle, USA, May

[2] O. Oliva, M. Faúndez "A comparative study of several ADPCM schemes with linear and nonlinear prediction" EUROSPEECH'99 , Budapest, Vol. 3, pp.1467-1470.

[3] M. Faúndez-Zanuy, "Nonlinear predictive models computation in ADPCM schemes". Vol. II, pp 813-816. EUSIPCO 2000, Tampere.

[4] N. S. Jayant and P. Noll "Digital Coding of Waveforms". Ed. Prentice Hall 1984.

[5] D. J. C. Mackay "Bayesian interpolation", Neural computation , Vol.4, Nº 3, pp.415-447, 1992.

[6] F. D. Foresee and M. T. Hagan, "Gauss-Newton approximation to Bayesian regularization", proceedings of the 1997 International Joint Conference on Neural Networks, pp.1930-1935, 1997.

[7] Chapter 7, Committee Machines of S. Haykin, "Neural nets. A comprehensive foundation", 2on edition. Ed. Prentice Hall 1999.

An Independent Component Analysis Evolution Based Method for Nonlinear Speech Processing

F.Rojas, C. G. Puntonet, I. Rojas, J. Ortega

Dpto. Arquitectura. y Tecnología de Computadores. University of Granada (Spain)
{frojas, carlos, ignacio, julio}@atc.ugr.es

Abstract. This paper proposes a novel Independent Component Analysis algorithm based on the use of genetic algorithms intended for its application to the field of non-linear speech processing. Independent Component Analysis (ICA) is a method for finding underlying factors from multidimensional statistical data, so it can be efficiently applied to suppress interferences and demodulate information in MultiInput-MuliOutput (MIMO) systems.

1 Introduction

The blind signal separation method such as Independent Component Analysis (ICA) is a novel nonlinear signal processing algorithm that performs linear data transformation that exploits high-order statistical information to project the data along the direction of maximal independence. The framework of the ICA is inherently analogous to that of multi-input multi-output (MIMO) systems. As a nonlinear signal processing technique, the ICA has demonstrated the enormous potential in many signal processing research areas. Viewing these facts, we expect that the ICA can be efficiently used to suppress interference and demodulate information data in MIMO wireless communication systems. Additionally, the replacement of a linear filter by a nonlinear model will guide us to more accurate results in real-world speech processing problems [6].

The guiding principle for ICA is statistical independence, meaning that the value of any of the components gives no information on the values of the other components. This method differs from other statistical approaches such as principal component analysis (PCA) and factor analysis precisely in the fact that is not a correlation-based transformation, but also reduces higher-order statistical dependencies. Therefore, blind source separation (BSS) consists in recovering unobserved signals from a known set of mixtures. The separation of independent sources from mixed observed data is a fundamental and challenging signal-processing problem [1], [3], [8].

Nonlinear ICA, on the other hand, is rather unconstrained, and normally demands additional information to make the estimations coincide with the estimations [9],[2],[11],[15]..

This paper is structured as follows: Section 2 introduces the post-nonlinear model as an alternative to the unconstrained pure nonlinear model. Afterwards, in Section 3, the basis of the algorithm is described: independence measure, probability density function estimation and evolutionary method depiction. Some experiments are shown in Section 4, using speech and synthetic signals. Finally, a few conclusion remarks and future lines of research terminate this paper.

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 679-686, 2003.

2 Nonlinear Independent Component Analysis.

2.1 Linear Model.

The ICA model when the mixture model is linear, i.e. a model which assumes the existence of n independent signals $s_1,\ldots, s_n$ and the observation of $x_1,\ldots, x_n$ instantaneous linear mixtures, can be represented by the following equation:

$$x(t) = A \cdot s(t). \tag{1}$$

where A is some unknown $n \times n$ matrix of real coefficients.

In the linear case we need to make the following tolerable assumptions, so the "blindness" of the method may be questioned [10]: the sources are statistically independent of one another, matrix A is assumed to be invertible and the sources have at most one Gaussian distribution.

Under this assumptions, we want to obtain a matrix W (separating matrix) whose output $y(t)$ would be an estimate of the sources $s(t)$:

$$y(t) = W \cdot x(t). \tag{2}$$

2.2 Post-Nonlinear Model.

The linear assumption is an approximation of nonlinear phenomena in many real world situations. Thus, the linear assumption may lead to incorrect solutions. Hence, researchers in BSS have started addressing the nonlinear mixing models, however a fundamental difficulty in nonlinear ICA is that it is highly nonunique without some extra constraints, therefore finding independent components does not lead us necessarily to the original sources [9].

Blind source separation in the nonlinear case is, in general, impossible. Taleb and Jutten [16] added some extra constraints to the nonlinear mixture so that the nonlinearities are independently applied in each channel after a linear mixture (see Fig.1). In this way, the indeterminacies are the same as for the basic linear instantaneous mixing model: invertible scaling and permutation.

The mixture model can be described by the following equation:

$$x(t) = f(A \cdot s(t)). \tag{3}$$

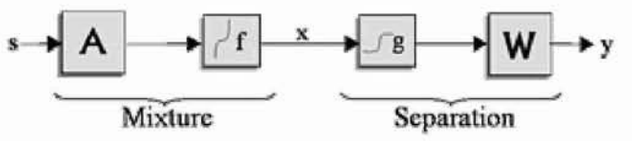


Fig. 1 Post-nonlinear model.

The unmixing stage, which will be performed by the algorithm here proposed is expressed by Equation 4:

$$y(t) = W \cdot g(x(t)) \tag{4}$$

The post-nonlinearity assumption is reasonable in many signal processing applications where the nonlinearities are introduced by sensors and preamplifiers, as usually happens in speech processing. In this case, the nonlinearity is assumed to be introduced by the signal acquisition system.

3 Algorithm Description.

3.1 Mutual Information and Entropy Approximation.

The proposed algorithm will be based on the estimation of mutual information, value which cancels out when the signals involved are independent. Mutual information I between the elements of a multidimensional variable $\mathbf{y}$ is defined as:

$$I(y_1, y_2, ..., y_n) = \sum_{i=1}^{n} H(y_i) - H(y_1, y_2, ..., y_n). \tag{5}$$

For Eq. 5, in the case that all components $y_1 \ldots y_n$ are independent, the joint entropy is equal to the sum of the marginal entropies. Therefore, mutual information will be zero. In the rest of the cases (not independent components), the sum of marginal entropies will be higher than the joint entropy, leading thus to a positive value of mutual information.

In order to exactly compute mutual information, we need also to calculate entropies, which likewise require knowing the analytical expression of the probability density function (PDF) which is generally not available in practical applications of speech processing. Thus, we propose to approximate densities through the discretization of the estimated signals building histograms and then calculate their joint and marginal entropies. In this way, we define a number of bins m that covers the selected estimation space and then we calculate how many points of the signal fall in each of the bins ($B_i \; i = 1,...,m$). Finally, we easily approximate marginal entropies using the following formula:

$$H(y) = -\sum_{i=1}^{n} p(y_i) \log_2 p(y_i) \approx -\sum_{j=1}^{m} \frac{Card(B_j(y))}{n} \log_2 \frac{Card(B_j(y))}{n}. \tag{6}$$

where *Card(B)* denotes cardinality of set B, n is the number of points of estimation y, and B_j is the set of points which fall in the j^{th} bin.

The same method can be applied for computing the joint entropies of all the estimated signals:

$$H(y_1,...,y_p) = \sum_{i=1}^{p} H(y_i | y_{i-1},...,y_1)$$
$$\approx -\sum_{i_1=1}^{m}\sum_{i_2=1}^{m}...\sum_{i_n=1}^{m} \frac{Card(B_{i_1 i_2...i_p}(y))}{n} \log_2 \frac{Card(B_{i_1 i_2...i_p}(y))}{n}. \quad (7)$$

where p is the number of components which need to be approximated and m is the number of bins in each dimension.

Therefore, substituting entropies in Eq.5 by approximations of Eqs.6 and 7, we obtain an approximation of mutual information (Eq. 8) which will reach its minimum value when the estimations are independent:

$$Est(I(\mathbf{y})) = \sum_{i=1}^{p} Est(H(y_i)) - Est(H(\mathbf{y})) =$$
$$= -\sum_{i=1}^{p}\left[\sum_{j=1}^{m} \frac{Card(B_j(y_i))}{n} \log_2 \frac{Card(B_j(y_i))}{n}\right] + ... \quad (8)$$
$$... + \sum_{i_1=1}^{m}\sum_{i_2=1}^{m}...\sum_{i_n=1}^{m} \frac{Card(B_{i_1 i_2...i_p}(y))}{n} \log_2 \frac{Card(B_{i_1 i_2...i_p}(y))}{n}.$$

where *Est(X)* stands for "estimation of x".

Next section describes an evolution based algorithm that minimizes the contrast function defined in Eq. 8, escaping from local minima.

3.2 Proposed Genetic Algorithm.

A genetic algorithm (GA) evaluates a population of possible solutions and generates a new one iteratively, with each successive population referred to as a generation. Given the current generation at iteration t, G(t), the GA generates a new generation, G(t+1), based on the previous generation, applying a set of genetic operations. Aside from other aspects regarding genetic algorithms, the key features that characterize a genetic algorithm are the encoding scheme and the evaluation or fitness function.

First of all, it should be recalled that the proposed algorithm needs to estimate two different mixtures (see Eq. 4): a family of nonlinearities g which approximates the inverse of the nonlinear mixtures f and a linear demixing matrix W which approximates the inverse of the linear mixture A. This linear demixing stage will be performed by the well-known FastICA algorithm by Hyvärinen and Oja [8]. To be precise, FastICA will be embedded into the genetic algorithm in order to approximate the linear mixture.

Therefore, the encoding scheme for the chromosome in the post-nonlinear mixture will be the coefficients of the odd polynomials which approximate the family of nonlinearities g .

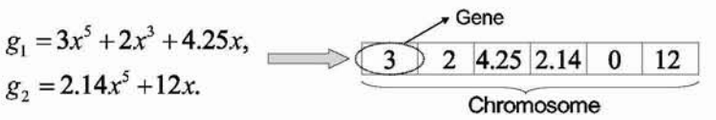


Fig. 2 Encoding sample for p=2 signals and polynomials up to grade 5.

The fitness function is easily derived from Eq. 8 which is precisely the inverse of the approximation of mutual information, so that the genetic algorithm maximizes the fitness function, which is more usual in evolution programs literature.

$$Fitness(\mathbf{y}) = \frac{1}{Est(I(\mathbf{y}))}. \tag{9}$$

Regarding other aspects of the genetic algorithm, the population (i.e. set of chromosomes) was initialized randomly within a known interval of search for the polynomial coefficients. The genetic operators involved were "Simple One-point Crossover" and "Non-Uniform Mutation" [13]. Selection strategy is elitist, keeping the best individual of a generation for the next one.

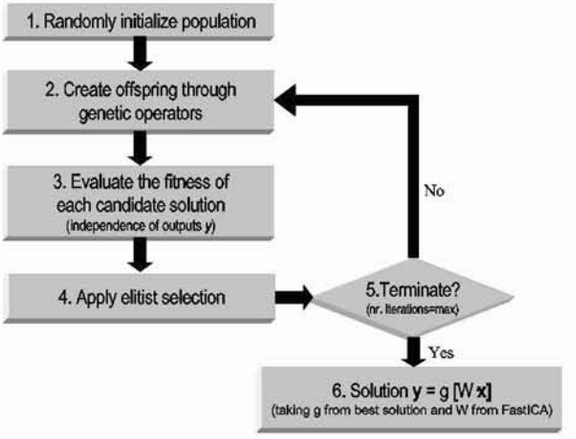


Fig. 3 Genetic algorithm scheme for post-nonlinear blind separation of sources

4 Speech Experiments.

This section illustrates the validity of the genetic algorithm here proposed and investigates the accuracy of the method. We combined voice signals and noise nonlinearly and then try to recover the original sources. In order to measure the accuracy of the algorithm, we evaluate it using the Mean Square Error (MSE) and the Crosstalk in decibels (Ct):

$$MSE_i = \frac{\sum_{t=1}^{N}(s_i(t)-y_i(t))^2}{N} \qquad \mathrm{Ct}_i = 10\log\left(\frac{\sum_{t=1}^{N}(s_i(t)-y_i(t))^2}{\sum_{t=1}^{N}(s_i(t))^2}\right) \tag{10}$$

4.1 Voice signal and a synthetic pulse.

A voice signal saying "Where are you?" and a square pulse were nonlinearly mixed according to the post-nonlinear model. The length of both signals was of 6912 samples. The following mixing matrix and nonlinearities were applied in order to obtain the nonlinear mixtures:

$$A = \begin{bmatrix} 0.74 & 0.87 \\ -0.9 & 0.14 \end{bmatrix}, f = \left[f_1(x) = \tanh(\frac{4x}{5}),\ f_2(x) = \tanh(5x) \right]. \tag{11}$$

Then the genetic algorithm was applied (population size=40, number of iterations=60). Polynomials of fifth order were used as the approximators for $g=f^{-1}$. Figure 4 shows how the approximations of the proposed solutions fit the original f^{-1}.

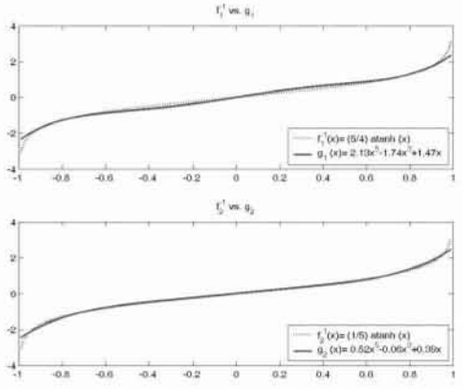


Fig. 4 Nonlinearities approximations compared to the original ones.

Performance indexes after performing normalization (zero mean, unit variance) to avoid dissonances due to invertible scalings and permutations, which are undetermined in the ICA model, where the following:

$\mathrm{MSE}(y_1, s_1) = 0.0014$ $\quad$ $\mathrm{Crosstalk}(y_1, s_1)$ (dB) = -18.94 dB,
$\mathrm{MSE}(y_2, s_2) = 0.0010$ $\quad$ $\mathrm{Crosstalk}(y_2, s_2)$ (dB) = -21.41 dB.

Figure 5 (left) shows the sources, mixtures and estimated signals, together with their scatter plots. As can be seen, estimations (**y**) are approximately equivalent to the original sources (**s**) up to invertible scalings and permutations. E.g. estimation y_1 is scaled and inverted in relation to s_1.

4.2 Two voice signals.

This experiment corresponds to the "Cocktail Problem" [4], that is, separating one voice from another. Two voice signals corresponding to two persons saying the numbers from one to ten in English and Spanish were non-linearly according to the following matrix and functions:

$$A = \begin{bmatrix} 1 & 0.87 \\ -0.9 & 0.14 \end{bmatrix}, f = \left[f_1(x) = f_2(x) = \tanh(x) \right]. \quad \textbf{(12)}$$

Genetic algorithm parameters were the same as those used in Sec. 4.1. Performance results and a plot of the original and estimated signals are briefly depicted below (Figure 5.b, right).

$MSE(y_1, s_1) = 0.0012$ $\quad$ $Crosstalk(y_1, s_1)$ (dB) = -17.32 dB,

$MSE(y_2, s_2) = 0.0009$ $\quad$ $Crosstalk(y_2, s_2)$ (dB) = -19.33 dB.

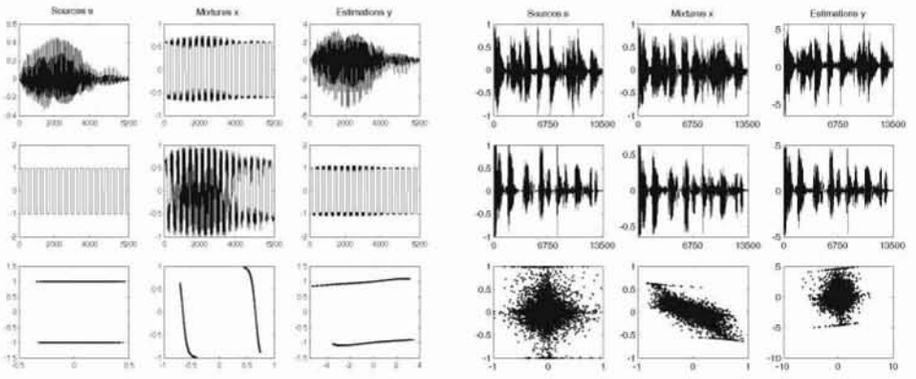


Fig. 5 Sources, mixtures and estimations (along time and scatter plots) for Section 4.1 (Fig 5.a, left) and Section 4.2 (Fig 5.b, right)

5 Concluding Remarks and Future Research.

In this work, the nonlinear speech processing problem has been tackled by an ICA algorithm based on the use of genetic algorithms. As the separation sources is impossible in nonlinear mixtures, we assumed a linear mixture followed by a nonlinear distortion in each channel which constraints the solution space. Experimental results showed promising results, although future research will focus on the adaptation of the algorithm for higher dimensionality and stronger nonlinearities.

Acknowledgement

This work has been partially supported by the CICYT Spanish Projects TIC2000-1348 and TIC2001-2845.

References

1. L. B. Almeida. ICA of linear and nonlinear mixtures based on mutual information. In Proc. 2001 Int. Joint Conf. on Neural Networks, Washington, D.C., 2001
2. G.Burel, Blind separation of sources: A nonlinear neural algorithm, Neural Networks, vol.5, pp.937-947, 1992.
3. J.F.Cardoso, Source separation using higher order moments, in Proc. ICASSP, Glasgow, U.K. May, pp.2109-2212, 1989.
4. E.C. Cherry, Some experiments in the recognition of speech, with one and two ears. Journal of the Acoustical Society of America, 25, 975-979. 1953.
5. P. Comon, Independent component analysis, a new concept?, Signal Processing, vol. 36, no. 3, pp. 287--314, 1994.
6. M. Faúndez, S. McLaughlin, A. Esposito, A. Hussain, J. Schoentgen, G. Kubin, W. B. Kleijn and P. Maragos, Nonlinear speech processing: overview and applications, in Int. J. Control Intelligent Syst., vol. 30, num. 1, pp. 1-10, 2002.
7. D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning,AddisonWesley, Reading, MA, 1989.
8. A.Hyvärinen and E.Oja, A fast fixed-point algorithm for independent component analysis. Neural Computation, 9 (7), pp.1483-1492, 1997.
9. A. Hyvärinen and P. Pajunen. Nonlinear Independent Component Analysis: Existence and Uniqueness results. Neural Networks 12(3): pp. 429-439, 1999.
10. C. Jutten and J.F. Cardoso. Source separation: really blind ? In *Proc. NOLTA*, pages 79-84, 1995.
11. T-W.Lee, B.Koehler, R.Orglmeister, Blind separation of nonlinear mixing models, In IEEE NNSP, pp.406-415, Florida, USA, 1997.
12. J.K.Lin, D.G.Grier, J.D.Cowan, Source separation and density estimation by faithful equivariant SOM, in Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, vol.9, 1997.
13. Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, New York USA, Third Edition, 1999.
14. P.Pajunen, A.Hyvarinen, J.Karhunen, Nonlinear blind source separation by self-organizing maps, in Progress in Neural Information Processing: Proc. IONIP'96, vol.2. New York, pp.1207-1210, 1996.
15. F. Rojas, I. Rojas, R.M. Clemente, C.G. Puntonet. Nonlinear Blind Source Separation using Genetic Algorithms, in Proceedings of the 3rd International Conference On Independent Component Analysis and Signal Separation (ICA2001). pp. 400-405, December 9-13, , San Diego, CA, (USA) , 2001.
16. A.Taleb, C.Jutten, Source Separation in Post-Nonlinear Mixtures, IEEE Transactions on Signal Processing, vol.47 no.10, pp.2807-2820, 1999.
17. D.Yellin, E.Weinstein, Multichannel signal separation: Methods and analysis, IEEE Trans. Signal Processing, vol.44, pp.106-118, 1996.

Ge e g Ge e A e e g

s l s t t^2 l

In o o c G N ro an o n or a c
n r o R g n rg D 0 0 R g n rg G r an
D rq c ra cno og a o a or
c a cn ca S r or Ing n r a In or a ca
n r a Grana a 0 Grana a

A st t In a r a g o r a a gor or non n ar n
o rc ara on r n x r ac co o n a
o conc n r c r ng n w c or nar n ar I r or n or r
o g a o ag o r ng o n n r or g na x ng a ng
ng o og r x ng a ng can r con r c ar
o a ca on o wo an r n ona ar c a an na ra a a
ar r n

n uc i n

d p d t p t l s s t s t t s d t
s t t *f* s s d p d t s p ss l s s ppl d t t l d s
s p t SS s w s l d t t s d l d
p d t s s *S*

l ss ll l SS w *AS* s t t d st t l
S l l l d s p t l t s d l p d st t w t
tt s l l t d t t l t
l t s d t w p p l t d
l l t s p p s d l pl l t s p st
l d s s d s l st l t st pp t
t d ls 6 p tt p ls pts s d s t
z t

p s t l z t t t sp l l d ls
s l t l t s s d t t t t w p
l l d st t t t d t t s ll t
t t t t t l w w ll s d d
t t sp d p l pp t s s d s
st d s t s d l t p s t d t t
t l d p d s t l t s d s l t d l
t t l d t ls t p s t d l t s z d

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 687-694, 2003.

2 Algorithm

We want to introduce a special kind of nonlinear symmetric ICA called **nonlinearGeo** that encompasses linear and postnonlinear ICA. The mixing mapping can hereby be a mapping f with $|f(x)| = |x|$ for all $x \in \mathbb{R}^n$. Indeed, we could also allow any increasing function in each coordinate, but for simplicity we will restrict ourselves to this case for now. Then $f(0) = 0$, and f maps centered random vectors to centered ones. So in spherical coordinates $(r, \Phi) := (r, \varphi, \vartheta_1, \ldots, \vartheta_{n-2})$, we have $f(r, \Phi) = (r, g(r, \Phi))$ for some possibly nonlinear function $g : \mathbb{R}_0^+ \times [0, 2\pi) \times [0, \pi)^{n-2} \to [0, 2\pi) \times [0, \pi)^{n-2}$.

Given a independent random variable $S : \Omega \to \mathbb{R}^n$, set as usual $X = f \circ S$. The basic idea of the presented algorithm then is to divide the mixture space $\mathbb{R}^n$ into a set of concentric rings with growing radii $r_1, \ldots, r_s$. In each of the rings so defined perform a linear ICA after central transformation in order to get $f(r_i e_j)$. These images of the basis vectors are then used to reconstruct the original signals. This idea is visualized in figure 1.

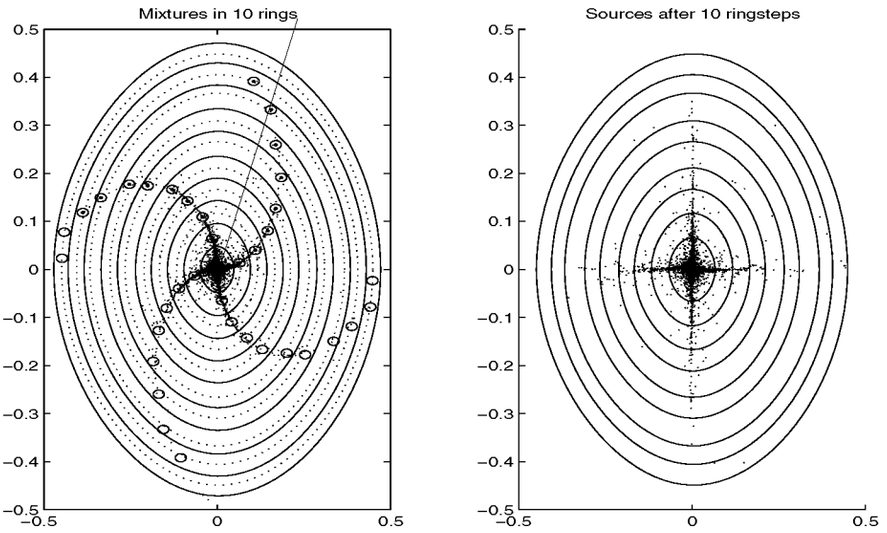


Fig. 1. Idea of nonlinearGeo: In the first step divide the mixture space in a set of equi-distant concentric circles, and perform linear ICA on the samples in each ring, left figure. The small circles indicate the images v_i^j of the recovered mixing matrices in each ring after normalization. In the second step, right figure, recover the sources.

The main algorithm consists of two parts:

- Mixing model recovery step: In the first step, quite similar to the BMMR step in overcomplete ICA [11], the mixing model is recovered. So given the samples X, choose $r_0 := 0 < r_1 < \ldots < r_s$. Denote $K_i := \{x \in \mathbb{R}^n | r_{i-1} \leq |x| <$

ss t t t s t t sp s
t ppl t l t s t γ R w p
l γ_1 d t t s A
l z t l s A w t t s v $\frac{1}{2}$
d t ddl d s $\frac{1}{2}$ d d

S st p s d st p w s s l t t S st p
pl t t s sts t s s s t st t d
s v t t s t s s l d ff t p ss l t s
st st d t t s w t p l t s t s pl t t
t s v v^n d t t t d s s w
w ll **ece e e r e r e e r e** s s l
d p p s d t t t s s t s t ls t p p l
t t pp s s p ts pl l
p p s d s st t t t l t s t t
s ll l s w t t ll l s
t s l t **r e r e** t t t st t t t t
pp s t s s s d pl l s s t p
l t w t p t pp d s pl t t t t
t d l t s d t ls t

t t t t l s t l l t d d ds t
l l d d s s s t s s lw s s lt t s
t d t st p t pl l d p l l
t s s s pl ts t s t s s l d
t d t ls t t l t d ts t t l d
p d d w p s t ppl t s d t p t ll t
t t t l s d t s p s t d

p i n su s

t s s t pl ppl t s l p s t d t
t s l t w s s l t t st st p t
l t l t st l t
pl w s d t tw sp s ls s t t
d s t l sp l pp

2 2

s s
s s

w ls s w t l s t s w d d
w p l l d t s s l s p s t
d ff t l t s t s pl w s t t p l l p s
t w ll ls s w t p l l t s w l l

s ws t t s t st l t s p t d p l l
p s st d t s t s l *S T*
w s s w l s t t s t p t t

gor	co ar anc	
o ar non n arG o	0 0 00 0 0 0	0 0 0
non n arG o	0 0 0 0 0 0	0
a G o	0 0 0 0 0 0	0
a I	0 0 0 0 0	0 0

xa r or anc r co r o non n ar x c gna
ro g r ng non n arG o an n ar a gor

s d pl w w t t s d p st l
pp w s s t

f t t 2
t 2

t tw sp s ls pl w t *f* s Us p l
l w s p t t s s ls t l t s
l s ws t p t s l t s p l l
p s t w ll s t t ls t s p st l s s l
d ll d l

gor	co ar anc	
o ar non n arG o	0 0 00 0 0 0	0 0
non n arG o	0 0 0 0 0 0	0
a G o	0 0 0 0 0 0	0 0
a I	0 0 0 0 0	0

2 xa r or anc r co r o o non n ar x c gna
ro g r ng non n arG o an n ar a gor

l st pl s ppl t l t d s l l
d s ls s s s w s t d st t d s ls w t

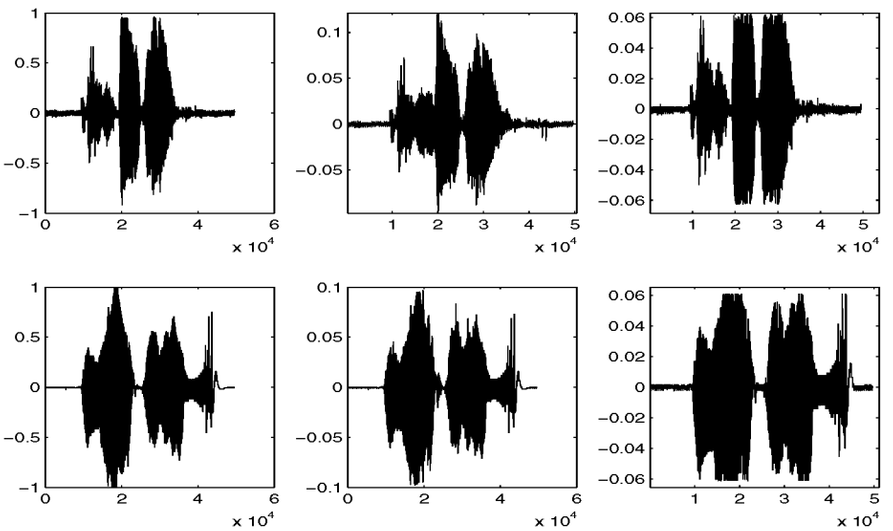


Fig. 2. Example 2: Nonlinear mixture of two speech signals. The left column shows the original sources, the middle column the mixtures, and the right column the recovered sources using polar nonlinearGeo.

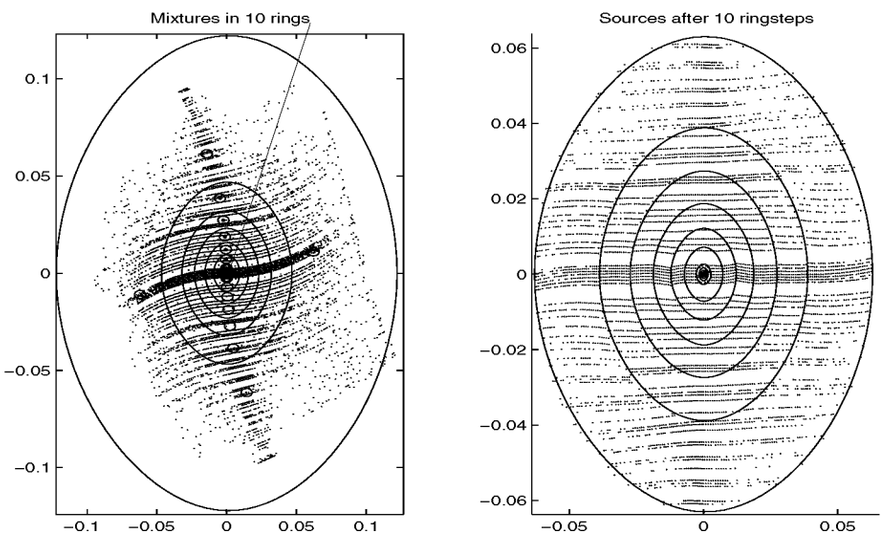


Fig. 3. Example 1: scatterplot of the nonlinear mixture from figure 2. To the left, we see a scatterplot of the mixture together with the equi-sampled ring divisions, and to the right, the scatterplot of the recovered sources using the polar inversion algorithm.

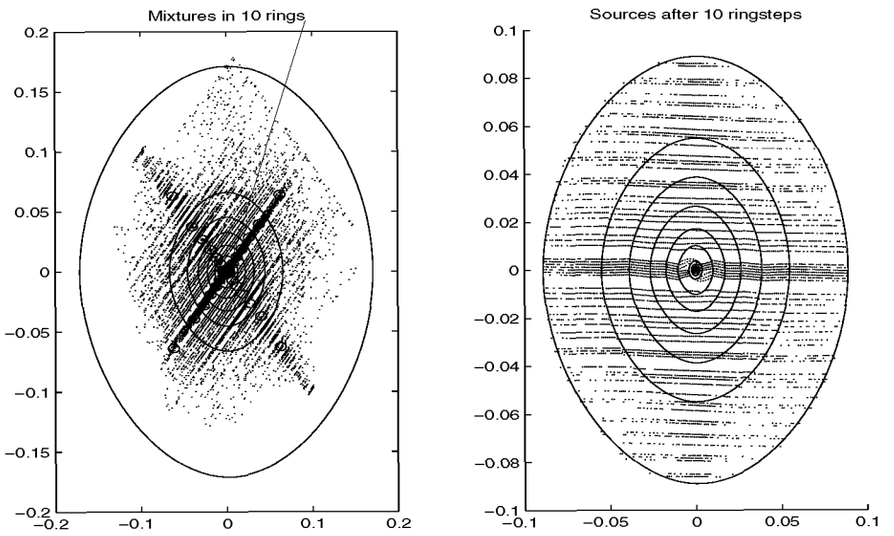


Fig. 4. Example 2: 2 speech signals mixed using the postnonlinearity f. Left figure shows scatter plot of the mixtures and right figure the one of the recovered sources.

kurtosis. They have been mixed using

$$\begin{aligned} \tau : \mathbb{R}^3 &\longrightarrow \mathbb{R}^3 \\ (x, y, z) &\longmapsto (\sigma(0.05x, 0.05y), 0.1z + \sigma(0.05x, 0.05y)_1) \end{aligned}$$

A scatter plot of the mixture is shown in the left-hand image of figure 5. There we also show the recovered sources using pl-nonlinearGeo with radii choosen such that the number of samples in each ring was constant.

The recovery error E_σ was only 0.195 which is very low in comparison to a linear recovery using fastICA, which produced a signal-recovery error of 0.984. As already the image visually confirmed, we can clearly see that we mixed the signals using a quite strong nonlinearity. Again nonlinearGeo performs well. In terms of implementation of the algorithm, we want to note that care has to be taken when fitting together recovered matrices from adjacent rings due to the scaling and permutation indeterminacies of ICA. As shown above, we account for scaling by assuming a polar mapping; hence we can use normalization in each ring. In order to deal with a permutation indeterminacy, we adjust adjacent matrices A_i and A_{i+1} by selecting a permutation matrix P such that $\|A_i - A_{i+1}P\|$ is minimal; then we replace A_{i+1} by $A_{i+1}P$. In two dimensions, this only means mapping closest angles to each other; in higher dimensions however, a list of permutations has to be generated and matrices adjusted as explained above. A similar idea is used in [13] in order to stitch matrices together after projections onto various hyper planes.

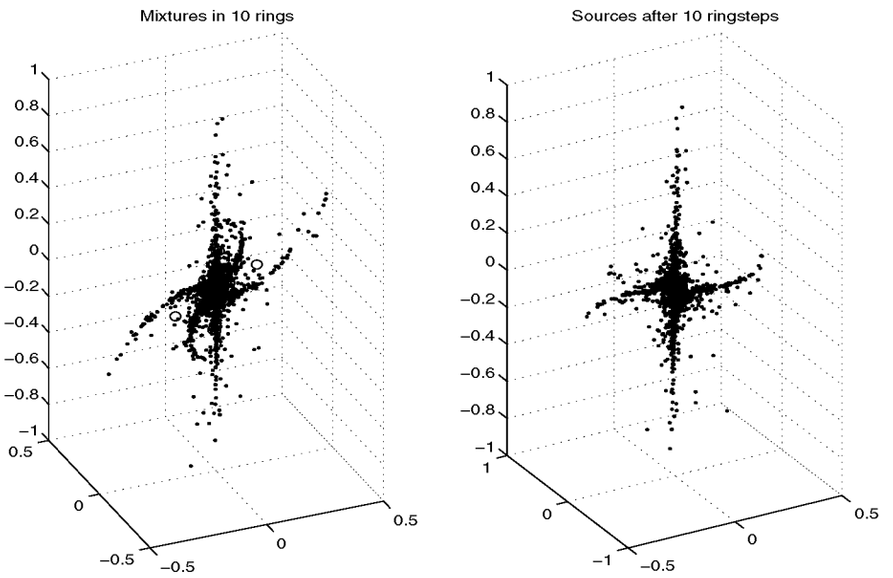


Fig. 5. Example 3: 3 dimensional example gamma distributed sources mixed using the nonlinearity τ. Again, the left figure shows the scatter plot of the mixtures and right figure the one of the recovered sources.

4 Conclusion

We have presented a new nonlinear BSS algorithm that can be interpreted as an extension of the linear geoICA algorithm. In the first step, we subdivide the mixture space into rings and find approximate linear unmixing mappings in each such region. In the next step, we use those approximated matrices to construct an unmixing mapping defined on the whole space; for this we recall Puntonet's piecewise-linear approach [9], and improve it using a so-called polar nonlinear approximation. We finish this chapter with some examples and see that the presented polar unmixing model even contains some postnonlinear problems.

For future work, we suggest studying the theoretical properties of the used model in more detail. It would be nice to directly get error thresholds for the approximation depending on the recovery algorithm. Furthermore other recovery algorithms like for example polynomial approximation using the v_i^j as points with given values could be used. First tests with polynomial approximations in two dimensions seem to restrict the degree to a maximum of 3 however, so that other approximation methods would still have to be applied afterwards.

References

1. J. Hérault and C. Jutten, "Space or time adaptive signal processing by neural network models," in *Neural Networks for Computing. Proceedings of the AIP Con-*

nc J S D n r N w or r can In o c
0
R n r n a ca on o r nc o ax n or a on r r a on
o n ar *anc n a n a n c ng Sy* o

a an J n So rc ara on n o non n ar x r
an n S gna c ng o 0 0
o r an R rg r n ara on o non n ar x ng
o *S* 0
a n n ar n n an J ar n n Non n ar n o rc ara on
organ ng a *g n a n a n c ng c*
n na na n nc n a n a n c ng H ng
ng o 0 0
J ar n n S a aro an I on oca n ar n n n co
on n ana a on c r ng *n a Sy* o 0 no
000
G arq an a S ara on o non n ar x r ng a rn
r on *a n ba n S*

J G n on an ang Non n ar g o r c I n *c*
n n 00
G n on R ar r o an r o S ara on o c
gna or non n ar x r *c n Sc nc* o 0 no
II
0 G n on an o r a r an ang S ara on o o rc ng
a ann a ng an co arn ng *c ng* o 00
J an ang or a a on o wo a roac o o rco
SS *c S* 0 00
J J ng G n on an ang n ar g o r c I
n a n a an a gor *a a n* o 00
J an ang ow o g n ra g o r c I o g r n
on *c S* 0 00

A Ad p e App B d e ep g e f g g p d e G

s l l z[2] l s t t[2] l

In o o c G N ro an o n or a c
n r o R g n rg D 0 0 R g n rg G r an
D rq c ra cno og a o a or
c a cn ca S r or Ing n r a In or a ca
n r a Grana a 0 Grana a

A st t n o rc ara on SS r o ran or a x
ran o c or n or r o r co r or g na n n n o rc
r n a n w a roac o n ar SS ng r a organ ng
a S or a n ra ga NG In co ar on o o r x r ac
ana g o r c a gor r n a con ra ro
n n ara on q a a o g co a ona co ra r
g n goa o a gor o a conn c on w n
n ra n wor an SS a co r r x o or xa
ran rr ng con rg nc roo or S o g o r c SS a gor

n uc i n

d p d t p t l s s t s t d st t st ll d p
d t d t w t d t ppl t l s l d
s s p t SS w t s t ss d t t t
s d s d s t d p d t s s d t ppl
l t s t SS p l s t st t l t s d l t s
s t t t t s t t tp t s ls s d p d t s p ss l
ls l d d st t st s
S t st t d t t t d lt d tt 6
s l t s p p s d t s l t l d s s p t p l

st t s d t t t t ls st s ll d
t l l t s t t s d t sp l s s w
st p p s d t t t l d t s
st d d d t l d d t s l t d w
t s lt d w st t l t ll d st w w ll
p p s tw w t l t s w l z t t sp s
s l z p sp t l l s

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 695-702, 2003.

CA n

l t t t t sp l t s d
l {*W* t d t *W* t l l p ^{n}
t { d t s t
t d t
d p d t d t *S* ^{n} w w ll ll d
rce vec r w t z d s t d st t w s d
p l t sp d *A* l s d t t l t w ll t
d l *A S* t **xe vec r** l l s t
t s s d t **x r x** *A* t t
t ll w w d t tw t s t t **e v e**
w tt s w t t l d l t
s l t l d t l t w t t t s
w p t t t l U ss l st t s t t t
st t s l s *S* s ss t **he**
y e r c B r e *D* l s t t *D* s
d p d t *D* s l t t *A* s t *D* l s
t t *D* s l t t *A* s l s t SS p l s w l l t t
t s d t l t

D *A* *A* *S*

t t t t t t s t d s l d p
t t d t s
t t l ss l t l t t s pl s d s
t sl t t d t t s ls *S* s t t t s d t
s s **ce ere** t ppl w t t s t t
t t s ll p p l p t l s s w l d d l t
t d t

ASS A A S A AA

w ss t t ls t s s w t d s s t t s l t
t l SS p l w ll ls s l t l SS p l s w
st t s l s t t s *A*

CA usin s f ni in p

e r z r h M s l st l t t
s d t s l z t d s l d t S s d l p d
d s t w d l s d d st d d
s l z t d l st t
t s s t w w t t d z t tw pts d S s
l d s t pp s t t s l l w
t t d t s st l st d s S d t s ppl d t

each cluster, or nonlinear BSS using a SOM as approximation to the demixing mapping [11]. Our approach is somewhat similar to Pajunen et al.'s idea [11] in the linear case but it does not require the sources to be subgaussian.

The idea of what we call **SOMICA** is very simple, based on the ideas of geometric ICA. Figure 1 shows the basic idea of SOMICA: Given observations X first whiten them such that $\mathrm{Cov}(X) = I$. Then use a SOM to approximate X. The corner unit locations then contain similar to geometric ICA the information of the mixing matrix A.

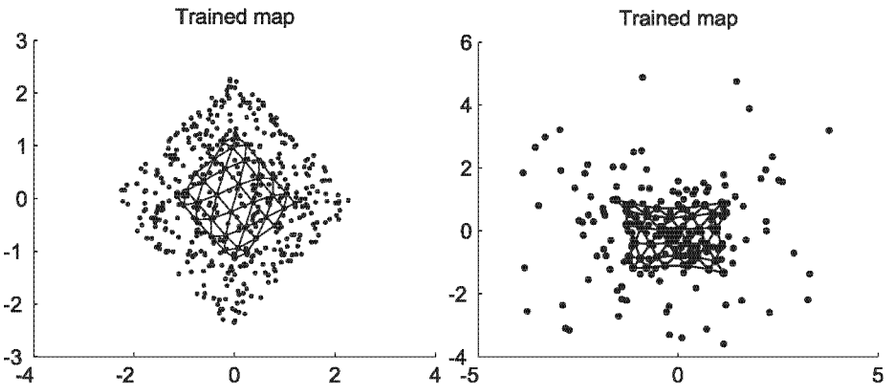


Fig. 1. SOMICA algorithm, sub(left)- and super(right)-gaussian case, Separation of a mixture of two uniform respectively Laplacian signals. The 2-dimensional SOM is used to approximate the whitened mixtures. The extremal units (those at the corners of the grid) are then images of the unit vectors or their sum, depending on super- or subgaussianity of the sources, so here $m_{11} = \lambda A(e_1 + e_2)$ (left image) and $m_{11} = \lambda A(e_1 - e_2)$ (right image) for some $\lambda \neq 0$. Crosstalking error of the separation in the subgaussian case was 0.108 and in the supergaussian case 0.0846.

So assume S is an independent non-gaussian 2-dimensional symmetric non-deterministic random vector, and let $X = AS$ with $A \in O(n)$ already whitened. Let $r \in \mathbb{N}$ and define a 2-dimensional SOM on the input grid

$$R = [1 : r] \times [1 : r].$$

Note that we index the processing units by 2-tuples $(i, j) \in R$. Use the SOM-learning-algorithm to approximate the whitened mixtures X. Let m_{ij} be the processing unit location of unit (i, j) after the learning process has converged. Define

$$B = (m_{11} - m_{rr} | m_{1r} - m_{r,1})$$

and

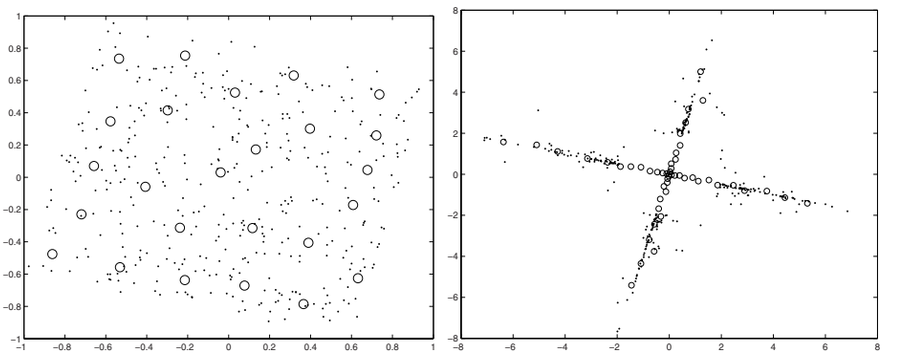
$$\hat{B} = (m_{11} + m_{1r} - m_{r1} - m_{rr} | m_{11} - m_{1r} + m_{r1} - m_{rr}).$$

l t t *S* s s p ss t s l t t *A* d t t *S* s s ss t ^ s l t t *A*
t t t w l t s w t t d
p p t l t *A* d *A* 2 *A* 2 d *A* sp t l s t t d *A* t s l s pt s l d p t t s t l t w ll t l t ll pl t t s l p s ld ll w t l s t t s d s t s lts S s
t t d w t s t s ld t s t t pl t s l s ss s t s
t S sp d t s t t d st t w d t d s *A* 2 S t t w ll t
l t t *A* Us s t w t s t ^ w t s t pp s t s d t st l z t l t t t s p ss s l t p t w t s t S s ld sp d d tl t *A* s w ll l t t *A* w s s st l t s s
S d s t w w t t w t s s t t t S l t l s p ts w l t t t pt lds w s t w w t t w s ld l p ts l ll t p ts l t t d t pt lds w t w t w t l st t s s d d t s
t t t t s t w t w t t s l t p t s s s p s ss t d t t t s l t p t t s d ^ d t t tt s l t t s l s l d s ppl d t tt l t w t t st t t t sp s pp t d s st

CA usin n u s

w w w t t s s l l t s s t t st d s S w w ll s l s
t **e r** st t d d t tz t l d s s d pt l s st w t w t t t w t d d t p t t z t t s tt t t l w pp t s t t t ds S l t S s sts s t s l t d p t sp **ce r** t t t w t sp d tp t t s d t l z sp d tw t p t sp d t tp t sp t s p p w s t l s l t pl t d t S l t ls p t st t S s t d s t t d l t s p w w w ll s t t t s s t ss d t s p t t d t

//www c / ro c / o oo ox/



F 2 NGI a gor an r r g ga an ca S ara on o a x r o wo n or r c ga a r gna r a ca r o o x r a n ona NG w r c 0 n o a rox a w n x r ga n x r a n ar ag o n c or or r n ng on r or ga an o o rc o r n ag or n r g ag or o 0

l t w w d t **I** w w s s l t t S l t d s s w t s d s p ss s w t d s t s w s l s t pp t U ts w t l d l s t t t t t s s

p t t tw d s l s w s l s w t l ts t t t p s t s [2] 2 d [4] w ss t t t d s t s s w t t d [2] pp s t t t s s t t d l s t s s l d ll d s p t t s t t s d d w p d tl s s t w t t s , d , t pl d [2] d [4] w l t t

[2] [4]

sp t l

^

[2] [4] [2] [4]

l t t *A* d p d w t *S* s s p s ss p t s l s st l t d t p t S l t s t s t s t t st d d s tt s t p ts S pp s t t t s s st t d s t S l t t sl t d t t l t d s p p p s w p t d t ls s t t w l z t d ff t pd t l s S d

d t l l l s t t t t t s s t l z t
d s s t ll l t p ts d t d
s t d pt d p l S l t s w st t d t
d s s

p s

t s s t w p S d w t t l t s
l t **I** l t d t tw t
l t s **e** d l pl t t **ceI**
d s s d l t w w d t **eI** tw

d s l s ls t $\sqrt{D}$ s d p d t d D

l l t d s t tl l d p s t l t t s l
w s p l s w t t s s p t d st t s d l
z d t s p p s w t S pl d
t s w w l t t s s d p p w t l l t s
p d w t d ws d tl s t S
l

p s w l l t t **er r ce ex cr**
err r s p p s d

n n n n

w A t l l t d st t A
st pl w s d t tw pl s ls s
s lts s w ll s t s t ll w t pl s s w t l
l t w s t l ps d p t p d t
sst l w t ts st d d d t st S pl
d t tw t l t s st S s sl w p s
t w t t d pt z t d S l t s s ll t d
t t sl w t s w S p s tt t
st s tw s st s S t s w t l ss t t
t t t s d t l p s t l
s l t
t s d d t d pl w p t s l t s
d d lt l d t d st t s t S d
t t sl w st d tt s t p l s w t t
d lt s w s t s p s s d t t t t t d lt d st t
s t d p d t t t l ss s p l t l t s d
s w t t t l t s l s d p l s w SS
d l s d d
t pl d ls w t l w ld d t tw d s ls tw sp
s ls s lts s l t t t pl s st tp s

S d t s sp d t st S d
st w p l ll w d S pl l t
s l ss t l d t t d ff t s d st t s

o ar on o r r n an cro a ng rror o I a gor or a
ran o x r o ar o gna an an an ar a on w r a n o r 00
r n w 000 a an n or r x ng a r x n

So rc	gor	/r n	n x
a ac an	a I ow		0 ±0
	a G o		0 ±0
	a c I		0±0
	S I		0 ±0
	S I		0 ±0
	NGI	0	0 ±0
n or	a I ow		0 0 ±0 0
	a G o		0 ±0
	a c I		0 ±0
	S I		0 ±0
	S I	0	0 0 ±0 0
	NGI	0	0 ±0
a r n c	a I ow		0 ±
	a G o		0 ±0
	a c I		0±
	S I		0 ±0
	S I		0 0 ±0 00
	NGI		0 0 ±0 0
o n c	a I ow		0 ±0
	a G o		0 0 ±0
	a c I		0 ±0
	S I		0 0±0
	S I		0 0±0
	NGI		0 0±0

C nc usi n

p s t d w pp l s l t t s
S d l s t t sp pp t t S d
st l d s lts p l sl tl tt
t t s t st l t d pt l t s t

t ts t p d st t sl w s t stl t st
t t l p t w sp ll t s t l z
d t t t l s lts l tw s t l t s
pl w p t p t t l t
s l t t S p d s
S l t s w t s t l d d l d st t s w ll
t p d s s t s t s p t
t t l t s ld t d d t t l s s l t

f nc s

S ar c oc an ang n w arn ng a gor or n gna
ara on *anc n a n a n c ng Sy*
J an J S now n n or a on ax a on a roac o n
ra on an n con o on *a a n*
J ar o o n gna ra on S a ca r nc a *c* 00
0
o on In n n co on n ana a n w conc *S gna c ng*

o r an J or n roc a o organ a on *nna*
n H n nca 0
J ra an J n S ac or a a gna roc ng n ra n
wor o In J S D n r or *a ng c ng*
n nc ag 0 N w or r can In o
c
ar n n an a a x o n a gor or n n n co on n
ana *a a n*
J ar n n S a aro an I on oca n ar n n n co on n
ana a on c r ng *n a Sy* 0 000
o o on n S organ ng or a on o o o og ca corr c a r a
yb
0 ar n S G r o c an J Sc n n ra ga n wor or c or
q an a on an a ca on o r r c on *an ac n n*
a
a n n ar n n an J ar n n Non n ar n o rc ara on
organ ng a *g n a n a n c ng c*
n na na n nc n a n a n c ng H ng
ng 0 0
G n on an r o n a a g o r ca roc r or n a
ra on o o rc *a c ng*
Ro r g ar Ro a G n on J r ga J an
ang g o r c I roc r a on a a c o o r a on ac
b 00
J J ng ang an G n on or c o or g o
r c n ar I *c 1* ag 00
J J ng G n on an ang n ar g o r c I n
a n a an a gor *a a n* 00

A A ed y e f e ge g

t s z d t z S t

D ar n o In or a on an o n ca on ng n r ng
n r o rc a S a n

A st t wor r n a o o og ca a roac o
r n or a on n n o wor w n c ac n
arn ng or c ca n ro c arc c r I
a o r co n a on w c a ow an r o wor g n r ca
w a a ag n ng ng ran ac on c r ng
a an xa o o a ca on o o o og ca
wor a o ow o n gra no on c r ng a gor
ro c o c n ro ro a a on ra a a
ow r w ag n ng oo can r ng a g n ra r o
oo or n c arn ng an w r ffor

n uc i n

w d s w d s tw s l t s s d tt z t
t d t ff d t s w t
d ff t s s t s l t t t s tt d sp s
t t s s s s t ll t d t l s s
t s tw d ff t p sp t s st s t t
d t s d s w s s w s s t s d
p sp t w t p p s w p s s t s w t
ts s t t s t d l
l t l t p l d t l t s s s t l s s tw t
t s s t d t s w ll ws t t s d d pl t
t d t l l t s d t sp d d t t s
t l t s p d sp ll w w ll s w s
s t p l s p s d l t s s l st st t t l
s z d s ll ws S t p s ts t w s l t s w
s t t t t s st S t s d t d t t d t
d t l pl t w lt w t t t t t s
s t t t st d s t t t t l t s d d t t s
t d ll w w ll p t t st p t t l s s t d
d s t w s s t

⋆ S or I ro c I 000 00 0

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 703-710, 2003.

A i s f w us inin

s s t ppl t d t t p tt s d s
w t s t s s s s l t p w ppl t s s
s p d t d ff t p s s p p ss p tt
d s d p tt l s s p s t w st d s p t l l
p tt d s l t s t p t l d w s
w w ll s t s d p s p ss p p s s d t
w t s t s d l t t d t s d p s l s s p s lt s
t d s d p tt s t s t t t t st t l
ppl t
ll d t l l t s w t t d t
t s l t d t t s w p s l st s d s ts s t
s s pl d t s s d s l t t t s s w d ff t
t t p w t d s t d s t l l s l st p t t p s
ll l st s d d s p ss s s d s t l l d s
t s w t d ff t ps d t t d w d t
st s pl s t s s t s s pl ss s t w p s st d
w s w s pl ss s l ttl t t
s d d tl l st ss s st *o* *d* st d t
t t s d t s ss s s ss s p ss s
t s s d t t s p t d t t d d
s ss s t d ff tw tw s t ss s d
t s s ss t t t
s d pl s ss s d l st l t w t
pl t l l t s t d d s ll t s l t s
t t d t pl t ll s s l t s
tw s ss s s s s d tw d ff t p s s s st
l t t d t d s , tw s ss s d t s t t t
l ss s tw t tw s ss s d d d t s ll ss s
ts s w t w s w t pl t d ll w t s s
t pl U s

/ u e / i t / u e / i e t

w ld t s s l t t

/ u e / i t / e /g t t

t t d t s s d s t t t t l pp
tw tw U s p t t t w ds w p t s l t s
t s d t t t t t s t t l ss s s s t d p
s s d t t t tt t t w
s d s l t t d t d s 2, s t t l s s d
s s s ll t st t w t t w *S* t ppl d s t s
d d pl U s d t d ff t s ss s s

S

w d t t t p t s t sp d d t s l
2, s ll t s l t s tw tw s ss s d
s

, 2,

t d w tw s ss s d ff t s t t ll ,
s ll t s d t l s t t d w t
tw s ss s s l t 2, s s s lts d t
l s d p d s ss t s ss s l s d ls
d ss l t s

2

s s l t z t p ss

n

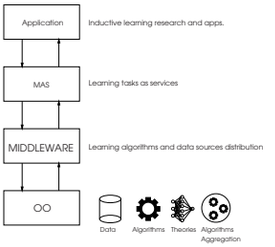
v

w s t s t t ds d d t sp p ss l s ss s s
t t t l s ss s c s t d l t s t t s ss v
s t t t d s t d ss l t s d s t s p
d t zz p s l st s ss t d v
tw st s s t l t t s tw t p l
s p t s d t t tw s s s p s l st s p
t s t s t s t t tl l s d l t t
p t t ff t t w l l t t t d t d ff t
s s t

M A A c i c u

s s t s tw d t s p d t d l l p
p t p t p l d t l t s s sp ll w
p d t l d t ld t l t s
s p s d l l d d t t l l tw s d
s t s d l s t l t s 6 d d t l ss t
t s s 6
ll w s s t s d t d t d s pt t t t
d t sp d p t t p s d l p d
s st t d t d ff t l ls l l s s
t l l pp p s t d t st d d p st l l s
t t t d l l t s s t U sp t s w d t
s s s t d t d t l l t s ts sp d
t d t s d t s t t l t s s d s t
ddl w l l t d st t s s s d ff t t ts
t d l l s lt t s st t s t l st l l t ppl t
st p t t t s t ll w

r **e** ts l t d t d ff t sts t ffi t
s l l s s
ex e **e** t s w t t d l t t t w d t l
l t s d t s w p d s l s t s t s
t s w
flex **e** ll ts t t ll d p d t d t s t ll ws
s d pt t t l st p t l t d t
c **e** t tl s pp t t w t p d l t s s t
p d l l s s t
t s p l d d t st w t d st t
l t s s t l l l s s s
er d ff t s s s d t l l t s
d t t s s t s st ff s t t s t ll ws
t s s d t d sp s ts w s sp t
d d w t p d d s lts



F r c r o arc c r

t d l l w l l t s d t s s
tw l t d pts s s d p s t t l s S s
s t t p t l st t l t s s t t t
p ss l t s sp w t t p p s d t st d t p s t t
p s t t l s s w t w t t s l t s sp s p s t d
pl s s p p t s d l t t t
w ts t l l tw s p t t p t t d t
t w t t s p t t p t t
p d p t sp s s ll s pp s t w l s d
s s s s p d t t p l p ss l
s l t s t s d s d s w t d s l t t t t t
d ff t t ff ts t t p t s l t s d
p s t t l s s t l s d l p pt pp t
s t s w t t l s ll s d s ss w
s d t ss t d p s t t l s d d ls p ss l
s l t s d p s t t pl s w

w t l tw s d s l s t ll l t s d t s
t t d t s w

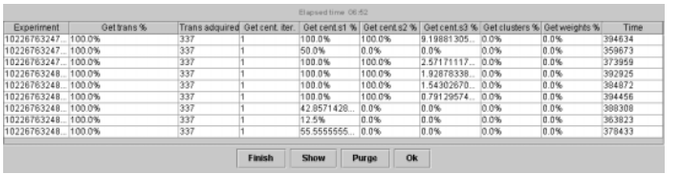
W us inin i s in i n

t t s w t t w l st l t s
l l s t tw sp ts t s l d st s t t
t t t w s p l d s d t p w t s p s d d t
l l st p st l t w s d d t ll s
l d s t s d d ls d ff t w s s p d
l st ss t l s d ls spl d d ls t
p w t s s t st t s s s d sp t s t t
w s s ppl t d p t l l t s t t
l st s w ll p s t l t s st t s s t t
s tw p t l z t d t t s d t l
t l t w t l
d t t t w l t t t s st t st t t d
s t d t d d ts t p t s s p t s t
s s s l t s t ll w

s pu s p t ll l l t d d t s
w tt t s t t d t w ll s d l t s s
s t st t w s l l s lw s d t t t t s
p t s ld lw s l st t s t 6
d_ d s l t t t lt w ss s ll ws t s t
d t w t st s lt d s s d d ll t s
ss s w t d ff t s lt d s t s d d
d_r s s t sts s d d s l d l pl
d PO
d_ s t s t st s t t d t t s
p t l d t p l d t t s s t t t t st t
l pl l s p p
d_d s sp t d ts t t t t t
sts l u . d s l s
u r t zz p t s d s s l t s
r s t t t l t t s t s l t
_ d s p t l t w l s ss s l l t
t t l t t d d t s t d
_ us r d t s t l st s l p t
r up p w l st d l t s d s s d t t d t
t s w t t U s s ss s ll U s s ss sp d
t t d ff t p t s d t t t s p t d t s t w
d pt w ll ll U s t t ll s d

ll t s p t s t d l p st p /s t ds
d t ll w t pl t t t t w l s

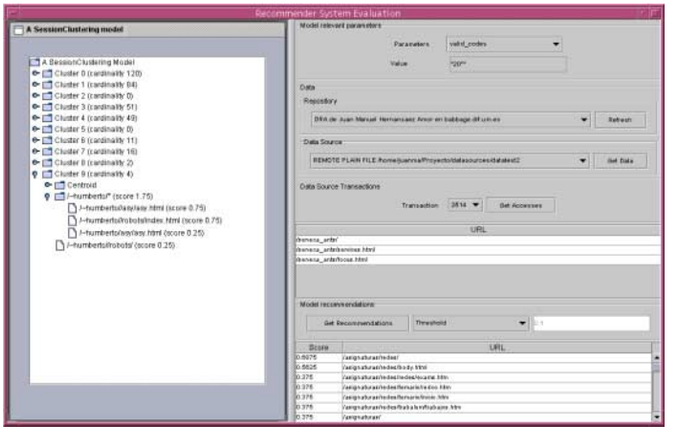
l t d d l t t d t U ll ws t d t w l t s s d t t w s l l t t d ss ppl d w d w w t s lts t s d t t st t w d w l t s s t t t U ll ws t d t lt l d p t l l pl t t p t _ us r t w ld t st w d t w d t ds d t t st p l p ss l d d t s t U w s t p ss l l s t t st l ts s t p t t U s ld t d ff t l t s d ff t l d t t ll p ll l ll l st d ls p d d w s l z d d d w t s t st s U ll ws t z t p ts t s t pl w st t d t w s ws t d



F 2 on or a on o arn ng o a r can o ow n rac ro g

t l t s s t d d ff t l st s l w p s ts l t s s d l t st t t l l w t s p t l s t t t d p ds s s t st l s ws t p t l ll d t t s d d t s t p ss t s p t t st p s s l t t t s ss s t d d t s s ss s t d t t p s s s t t w s l l s d d t s ss s w t d s l s t s ll ws s ll l t s s s t s d t t l s t t t s p d t s l t ll w t d t t p ss t t p s s t s l t t s ws t p ss t l st s t ds t d t s t t t t ss s t U l st l st l s t t t ll s ds l ps d t ll l s s d w s t s l z t s t s t w l st d l t d t w d l p d t s d d l s t t l t w d w t t w s t st t w s ws l st d U d t l st w d t U s d t p t s w p d t l st t t p t d t p t tt w s t l st p t s d l s

t l t s d t l l d w t w d t w s t s t s d t s t s w t t l st d l t t t p t t w d w w s t l t l t t s d l t s t d w U s t s ld t st d s t t U s t s tl s t s ss s t d t d t s w s l t t l l d w t r s w s t U s t s ss d tl l w d ll w t s t d t s t d l s t t U s s st d w s l t r s d s t s st d U s t pp



F D a r c or a c rng o o a n ro a w r r og

C nc usi ns n fu u w

ff s t d l l pp t ld t s st s s d s p s d d s p s d l s t s d st t d w t t t t l l st l t s t d d t w s t l t d l l t t d t l l t s s p d t s l ld p w l w s t l w t l ttl ff t s p s l p p s d t l s st t t t w d ls d ll s s t d t l st d ls t d U s t s t w w tl d l p w s t t w s l t t s p p s s t w ls t st d t t t l st d ls t w s s t p d w t d s st d t st t U s t w p s ll w w t t

t d t t t w l st l t st t s s s
s d s p t d p t ss l st s

f nc s

r o r o *a* *a n c gn n* ar n on r
x or
J an o a *n c n n n n b a a n za a*
c na n c a c a D D o Ing n r a a
In or ac on a o n cac on n r a rc a 00
J an o a n on o S ar a rc Gar o an J an R a co an
ng a arg n r o ac n arn ng x r n n a a a In
n g n Sy n a c an Sca ab y n ag n
Sy n g n n nc on r a J 00
R oo o a r an J Sr a a a Da a r ara on or n ng wor w
w row ng a rn *na n g an n a n Sy*
R D an Ja n *g a a a* r n c a ng woo
ff
Da Go rg *G n c g n a c za n an ac n a n*
ng on
o r an J S Da a n ng arc or now g n
a a a c n ca R or S R 0 n r oor n n In or
a ca o r Sc nc /D ar n o gor c an rc c r
n a Jo an Rag r na ra n n ng w acc og In
S G n a c n a a n ng an n g c y
ag 000
Ra on o a a an n r oc n ng r arc a r In
S G a n J 000
0 R r na a an N ana or *a ca n a n c gn n an*
c n n na y o o *Han b S a c* Nor o an
r a
a a o a r Ro r oo an Ja Sr a a a o a c r ona
a on a on ag n ng *n ca n*
000
Na rao r g Jo an R r na ra n ng w acc og
ng r a ona co c r ng g
Jo n o r ro o an r c G c anan In c o c rag a c o
a c on *ac n a n ng* 0
J Ro n an *g a ac n a n ng* organ a ann
r n ac n arn ng organ a ff an San a o a orn a
Ja Sr a a a Ro r oo n D an an ang N ng an
ag n ng D co r an a ca on o ag a rn ro w a a
S G a n 000
Ian n an ran *a a n ng ac ca ac n a n ng*
an c n n a n organ a ff an 000

Web Meta-search using Unsupervised Neural Networks

Sergio Bermejo & Javier Dalmau

Department of Electronic Engineering,
Universitat Politècnica de Catalunya (UPC),
Jordi Girona 1-3, C4 building, 08034 Barcelona, Spain
e-mail: sbermejo@eel.upc.es

Abstract. Web mining aims to learn regularities automatically in the World Wide Web for retrieving useful information. In spite of the enormous potential of soft computing techniques like neural networks (NN) for web mining, their use has been very restricted to date. Our work examines and discusses the application of unsupervised NN to group retrieval results in a novel meta-searcher.

1. Introduction

A significant increase in the amount of information accessible on the World Wide Web (WWW) has occurred in the last few years. Search engines like Google [1] are faced to index more than 1000 million pages and their users are interested in retrieving useful information (e.g. links to Web pages and documents) according to their interests. In this context, techniques that automatically discover and extract information from the WWW, which has been given the name Web mining [2], will enhance the quality of the retrieved search results. Web mining is based on the use of data mining methods [3][4] that are placed in the common frontiers of different fields like databases, artificial intelligence, machine and statistical learning, pattern recognition and data visualization. Recently, the potential of soft computing techniques like neural networks (NN) for enhancing current web mining applications [5] has been highlighted, since their use in this area has been very limited so far.

In this paper, we will explore the use of unsupervised NN in a web mining area of enormous interest: search engines. In particular, we will apply the popular k-means clustering algorithm to organize retrieval results from a novel meta-searcher. The rest of the paper is organized as follows. Section 2 reviews the application of data mining in search engines and considers several limitations of current approaches. Section 3 discusses the role of clustering in meta-search engines. Section 4 introduces a novel meta-searcher that groups web links with k-means and presents some preliminary experiments. Finally, Section 5 presents some conclusions and suggestions for further work.

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 711-718, 2003.

2. Web Mining in Search Engines: Prospects and Problems

2.1. Web Mining Elements and Problems

As suggested by several authors [5][6], Web mining problems, like other related information-processing problems, can usually be broken down into four sub-tasks:

1. **Information Retrieval (IR)**: An automatic retrieval of relevant documents is performed, ensuring that non-relevant ones are filtered as much as possible.
2. **Information Feature Selection/Extraction**: Once the documents have been obtained, some specific features of the document (i.e. index terms) are extracted and selected in order to facilitate the processing of further stages.
3. **Machine learning**: Empirical mapping concerning the goal of the processing information system (e.g. a user profiler or a grouper of Web documents) is built upon the selected features. Unsupervised learning methods must play a predominant role since much of the Web data is not labeled. Also, good generalization performance, i.e. the response to unseen data during training, is mandatory in order to obtain a reliable mapping to be used in the next stage for analysis purposes.
4. **Post-analysis**: The data available from machine learning is analyzed in order to understand, visualize and interpret it.

Web mining applications must face many challenges, which in many cases originate from the nature of the Web data itself. Among the Web data problems [19], we particularly stress the following:

1. Data is widely distributed across a large network of interconnected computers with no predefined topology and with a highly variable bandwidth.
2. Data is highly volatile since many computers and data can be added or removed (In 1999, 40% of the WWW was estimated to change every month).
3. Data is sometimes hidden since it is generated dynamically.
4. Data is often unstructured and redundant and can include many errors of different origins (old-fashioned, poorly-written, grammatical mistakes, etc.).
5. Data grows exponentially, so numerous scaling problems arise for coping with a large amount of information

2.2. Searching the Web

There are essentially three different ways of searching the Web that are widely employed in commercial search engines. The first is to rank a portion of Web documents found in the database of the search engine, formerly updated by a web spider included in the engine which crawls the Web, according to the search terms introduced by the user. The second, closely related to the first one, is to exploit hyperlinks, i.e. links found on a Web page, to refine previous searches. And finally,

the third is to use Web directories that organize Web documents by subject, which is known as browsing. Current practices are focused on supplying a combination of these solutions.

Most users go through search engines to obtain specific information available on the Web, which in some cases is ordered according to an estimated relevance. Additionally, links to similar pages are also provided as a search result. While search engines offer large Web coverage (millions of Web pages), browsing in Web directories of search engines is restricted only to a reduced spectrum of the WWW (estimated at being less than 1% of all Web pages). However, the documents provided to the user by Web directories are often much more significant. Also, the current instability observed in major WWW search engines [9], possibly due to quality-for-speed trade-offs made by them, considerably limits the quality of the information retrieved.

2.3. Meta-searchers

One simple idea for obtaining stable information from the results of single search services, which vary considerably according to the time and the engine, is to merge the answers from several search engines into a single one. With this approach, low-quality links can be identified and eliminated quickly [10]. Through a parallel search in several search engines, a meta-search softbot like MetaCrawler [11] allows significant documents to be obtained by merging and ranking the retrieval results of single queries. A meta-searcher must accomplish some specific tasks related to its meta-service, like the understanding of queries and output formats supported by each search service and the elimination of duplicated information. The way in which meta-searchers can retrieve information on search engines offers many possibilities. The most direct is a query based on search terms, but browsing in directories or exploiting dynamic search methods [12][13] based on following links in retrieved documents is also a good option.

3 Clustering in Web Searching

3.1. Clusters and Categories

A quite unusual approach for organizing retrieved documents in current search engines is document clustering, which aims to form natural groups of similar documents [6][7]. However, this method could help the user to gain a deeper understanding of how the recovered collection is organized. In order to apply clustering (or any other machine learning algorithm) to text processing, each document must be represented as a feature vector that numerically represents the information included in it [14]. Additionally, a clustering algorithm employed in Web search must fulfill some key requirements [6]:

1. **Finding coherent clusters**: The algorithm must find natural groups that contain similar documents.
2. **Extraction of relevant group information**: The algorithm must provide short and precise descriptions of each natural group that summarize the information contained in the documents.
3. **High speed**: The algorithm must be able to cluster thousands of documents in a few seconds.

Although text clustering seems to identify heterogeneous collections of documents and help to discover meaningful themes, in some cases a high variability of results among different queries has been observed [7], which is related to its unsupervised approach to learning. In this context, the use of categories from Web directories can help to find more stable clusters that can bias the learner with supervised data. A related approach can be found in the intelligent browsing of Intranets, in which certain domain knowledge is applied to retrieve more significant information [18].

3.2. Clustering with K-means

One form of clustering is achieved through vector quantization [15], which aims to design a vector quantizer VQ of dimension p and size K, i.e. a mapping from a p-dimensional Euclidean space, $\Re^p$, into a set or codebook $\mathbf{C}=\{\mathbf{w}_j,\ j=1,...,K\}$. Associated with every code vector $\mathbf{w}_j$ there is a region of influence R_j (or cluster) where VQ maps any input vector that falls into it to $\mathbf{w}_j$. The simplest form of a VQ is based on a nearest-neighbor rule in which R_j is defined by

$$R_j = \left\{ \mathbf{x} \left| \left\| \mathbf{x}-\mathbf{w}_j \right\| = \min_{i=1,...,K} \left\| \mathbf{x}-\mathbf{w}_i \right\| \right. \right\} \tag{1}$$

Thus, $VQ(\mathbf{x})$ can be expressed as

$$VQ(\mathbf{x}) = \sum_{j=1}^{K} 1(\mathbf{x} \in R_j)\mathbf{w}_j \quad \text{where } 1(\mathbf{x} \in R_j) = \begin{cases} 1 & \text{if } \mathbf{x} \in R_j \\ 0 & \text{if } \mathbf{x} \notin R_j \end{cases} \tag{2}$$

and then the codebook $\mathbf{C}$ is computed through the minimization of a measure of performance considering the total sequence of input patterns to be quantized, which can be expressed in terms of a statistical criterion. In optimal K-means, the statistical criterion is the expected error of the vector quantizer given by the functional

$$I[VQ] = \frac{1}{2} \int_{\Re^p} \sum_{j=1}^{K} 1(\mathbf{x} \in R_j) \left\| \mathbf{x}-\mathbf{w}_j \right\|^2 \Phi(\mathbf{x}) d\mathbf{x} \tag{3}$$

where $\Phi(\mathbf{x})$ is the probability density function of X. Since $\Phi(\mathbf{x})$ is often unknown, we build an empirical estimator of I[VQ], $I_{emp}[VQ]$ with a set of random vectors in the following way:

$$I_{emp}[VQ] = \frac{1}{2N}\sum_{j=0}^{N-1}\sum_{i=1}^{K} 1(\mathbf{x}_j \in R_i)\|\mathbf{x}_j - \mathbf{w}_i\|^2 \quad (4)$$

where the samples $\{\mathbf{x}_i, i=0,\ldots,N-1\}$ are known as the training set **T**. Batch K-means make use of the Newton optimization method over $I_{cmp}[VQ]$, which is defined as:

$$\mathbf{W}[n+1] = \begin{bmatrix} \mathbf{w}_1[n+1] \\ \vdots \\ \mathbf{w}_K[n+1] \end{bmatrix} = \mathbf{W}[n] - \mathbf{H}[n]^{-1}\frac{\partial I_{emp}[VQ[n]]}{\partial \mathbf{W}[n]}, \quad n \geq 0 \quad (5)$$

with **H** denoting the Hessian matrix H. As [16] shows, the learning equation is simply

$$\mathbf{w}_j[n+1] = \frac{1}{N_j[n]}\sum_{i=0}^{N-1} 1(\mathbf{x}_i \in R_j[n])\,\mathbf{x}_i, \quad n \geq 0, \quad j = 1,\ldots,K \quad (6)$$

where N_j is the number of training samples that fall into the Voronoi region R_j. Two additional remarks are worth noting. First, the generalization error of K-means for finite sample sizes can be bounded using tools of statistical learning [17], which allows practical measures of performance to be obtained. Furthermore, the use of Newton optimization at a much-reduced computational cost (since the Hessian matrix is diagonal) allows batch k-means to be applied in real-time problems like web search.

4 Intelligent Meta-searchers: A Case Study

4.1. Architecture

Figure 1 shows the architecture of our meta-searcher. A search agent is responsible for submitting user queries to the search engines Yahoo and Metacrawler (a meta-searcher). The idea of combining Yahoo and Metacrawler is to merge the results of meta-searching with Web directories provided by Yahoo. Each retrieved result is stored in our database, and any repetition is eliminated. Then, the filtering agents extract a feature vector of 21 components for each result with the frequency of the most common index terms found in the search. Furthermore, a spider provided to explore a particular domain and fish and shark searches [12][13] from the retrieved documents can also be activated. Finally, a clustering agent groups the results into categories, as described in Section 4.2. The user interface of our meta-searcher is shown in Figure 2.

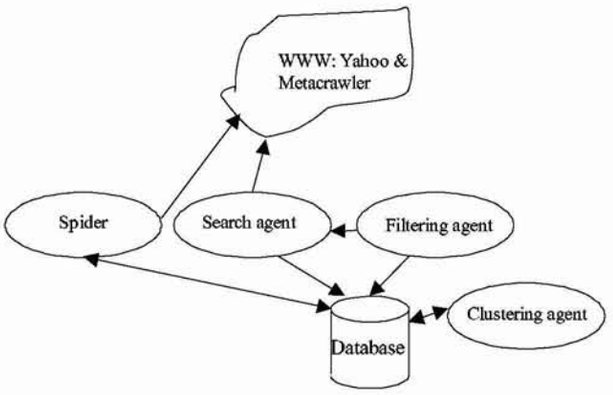


Fig. 1. Architecture of our metasearcher.

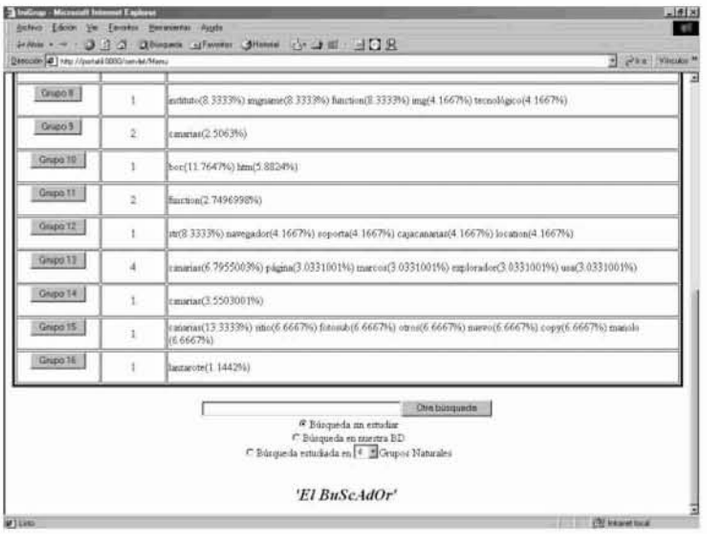


Fig. 2. User interface of our k-means meta-searcher.

4.2. The K-means Meta-Search Algorithm

The aim of our algorithm is to group N search results obtained from one or more search engines, in which each result is a text chain, into K natural groups in which K can be automatically estimated. The adaptive meta-search algorithm is divided into three steps:

1. **Preprocessing of search results**: which obtain the keywords or index terms for each retrieved document. This processing is divided into the followings steps [19]:
    a. *Lexical analysis*, which converts a stream of characters into a stream of words
    b. *Elimination of stopwords* with the aim of filtering words with no meaning for retrieval purposes.
    c. *Stemming* or affix elimination
    d. *Index selection,* which determines the words employed as indexing elements.
2. **Feature extraction**: Once the preprocessing is performed on N retrieved results $\{B_j, j=1,...,N\}$, an index vector $\mathbf{b_j}=\{b_{j1}, b_{j2}, ..., b_{jQ}\}$ can be formed where $b_{ji}=1$ if B_j contains the key word I_i from the set of indexing terms $\{I_i, i=1,...,Q\}$, with Q denoting the number of existing indexes. However, the use of a binary indicator does not take account of the importance of each index term in the document. Hence, a feature component b_{ji} that denotes the frequency of occurrence of the index term I_i in B_j is preferable.
3. **Training**: The k-means clustering algorithm computes the natural groups according to the set of features vectors obtained in preprocessing. Using a validation set, i.e. a subset of the original data, estimation of the optimal number of training epochs and clusters is also feasible.

4.3. Empirical evaluation

Three key terms were investigated with our intelligent search engine: "salamandra", "learning", "neural networks". The k-means meta-searcher detected five relevant natural groups. Table 1 shows the results. As one can observe, an automatic detection of meaningful categories was achieved. However, several overlaps of categories were detected in the formed natural groups. For instance, one of the salamandra groups is composed of pages about people and companies named Salamandra as well as Spanish pages about the amphibious Salamander. The same happens in "learning" groups in which machine learning and e-learning pages are collapsed into one single group. Additionally, pages that does not apply to the extracted categories were detected in the all the natural groups. All these limitations are in fact related to the nature of the learning process which was unsupervised.

Key term / Group	Salamandra	Learning	Neural Networks
1	Food company (2)	Machine learning (100) + e-learning (100)	survcys (4)
2	People and Companies (20)	Teaching→Courses (7)	NN in spanish (13)
3	Amphibious Salamander (9)	Teaching→Courses (5)	Publications and conferences (75)
4	People and companies (35) + Amphibious Salamander(5)	Machine learning→courses (40)	Research groups (14)
5	Web domain (2)	Machine learning→ researchers (10)	Researchers (25)

Table 1. Natural groups automatically detected with K-means meta search.

5. Conclusions

Web mining emerges as one of the most interesting applications of machine learning techniques like neural networks. However, it creates many challenging problems since web data is large, redundant, distributed, and volatile. Additionally, it often contains many errors that cause learning algorithms to fail. In our work, we have explored the application of unsupervised learning methods for the clustering of Web documents in a meta-search engine. In spite of the promising results, further work must be done to obtain more homogenous clusters through a robust feature extraction process and also to group documents within a hierarchy according to pre-established (or on-line) categories.

References

[1] Google search engine. http://www.google.com/
[2] Etzioni, O. (1996). The World Wide Web: quagmine or gold mine? Communications of the ACM, Vol. 39, no. 11, pp. 65-68.
[3] Adriaans, P. & Zantinge, D. (1996). Data Mining. Reading, MA: Addison-Wesley.
[4] Witten, I. H. & Frank, Eibe. (1999). Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Boston, MA: Morgan Kaufmann.
[5] Pal, S.K., Talwar, V., Mitra, P. (2002). Web Mining in Soft Computing Framework: Relevance, State of the Art and Future Directions. IEEE Transaction on Neural Networks, Vol. 13, No. 5, September 2002.
[6] Zamir, O. & Etzioni, O. (1998). Grouper: A Dynamic Clustering Interface to Web Search Results.
[7] Hearst, M. (1999). The use of categories and clusters for organizing retrieval results, in Strzalkowski, T. (Ed.) Natural Language Information Retrieval. Norwell, MA: Kluwer Academic Publishers.
[8] Nick, X. Z. & Themis, P. (2001). Web search using a Genetic Algorithm. IEEE Internet Computing, March-April, pp. 18-26.
[9] Selberg, E. & Etzioni, O. (2000). On the instability of Web Search Engines. Available in www.cs.washington.edu/research/clustering
[10] Selberg, E. & Etzioni, O. (1995). Multi-Service Search and Comparison Using the MetaCrawler. Proceedings of the 1995 World Wide Web Conference. Available in www.cs.washington.edu/research/
[11] Knoblock, C.A., Mauldin, M. L., Selberg, E., Etzioni, O. (1997). Searching the World Wide Web. IEEE Expert, January/February, pp. 8-14.
[12] De Bra, P. M. E. & Post, R. D. J. (1994). Searching for arbitrary information in the WWW : The fish search for Mosaic. Proceedings of the 1994 World Wide Web Conference.
[13] Hersovici, M. et al. (1998). The shark search algorithm. Proceedings of the 1998 World Wide Web Conference.
[14] Mladenic, D. (1999). Text-learning and Related Intelligent Agents: A Survey. IEEE Intelligent Systems, July/August, pp. 44-54.
[15] Gersho, A. & Gray, R.M. (1992). Vector Quantization and Signal Compression. Boston, MA: Kluwer Academic Publishers.
[16] Bottou, L. & Bengio, Y. (1995). Convergence Properties of k-means, in: Advances in Neural Processing Systems 7. Boston, MA: MIT Press.
[17] Buhmann, J. M. & Tishby, N. (1998). Empirical Risk Approximation: A Statistical Learning Theory of Data Clustering, in Bishop, C. M. (Ed.) Neural Networks and Machine Learning, Berlin: Springer-Verlag.
[18] O'Leary, D. E. (1997). The Internet, Intranets, and the AI Renaissance. IEEE Computer, January, pp. 71-78.
[19] Baeza-Yates, R. & Ribeiro-Neto, B. (1999). Modern Information Retrieval. New York: ACM Press.

Virtual Labs for Neural Networks E-courses

Sergio Bermejo, Ferran Revilla and Joan Cabestany

Department of Electronic Engineering, Universitat Politècnica de Catalunya (UPC)
Jordi Girona 1-3, edifici C4, 08034 Barcelona, Spain
sbermejo@eel.upc.es

Abstract. Human learning over the Internet (e-learning) aims to improve both the availability of information and the performance of the students involved. Virtual laboratories are one form of e-learning in which students learn practical skills by carrying out practical work. Our work here discusses the impact of e-learning, virtual labs and intelligent tutoring systems in education. We also introduce a neural network e-lab, which has several features that support active learning and assist the assessment of the students.

1 Introduction

The use of information and telecommunication technology (ITT) in higher education is allowing for the progressive disappearance of the limitations of space and time. ITT systems have many features that promote the active, experimenting and investigating student [1]. Some of the main changes that ITT systems will induce are the development of students' competencies and practical skills in virtual laboratories.

This paper introduces a virtual laboratory of neural networks (NNs) for Internet courses based on personal assistants, which promotes the active role of students in their learning and provides several features that facilitate their assessment. The rest of the paper is organized as follows: Section 2 introduces the basics of e-learning, describes how virtual labs are organized, and briefly reviews intelligent tutoring systems (ITSs), with a particular emphasis on personal assistants; and Section 3 introduces a virtual NN lab and its basic features. Finally, several conclusions are drawn and further work is suggested.

2 E-learning and Artificial Intelligence

2.1 E-learning

Electronic learning, or e-learning, refers to the use of Internet technologies to deliver a wide spectrum of learning solutions with the aim of improving the availability of knowledge and the performance of the students involved. It is based on three main

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 719-725, 2003.

principles [2]: networking; delivery through standard Internet technologies; and a view of learning that goes beyond the teacher-student paradigm, with an increased emphasis on informal and on-demand learning [3][4][5]. Generally speaking, the use of the Internet in this way has a wide range of potential advantages, such as [1]:

(1) **The availability of digital format contents**. Information can be instantly accessed, stored, distributed, shared and updated.
(2) **Support for group work**. Computer networking facilitates associative and group work.
(3) **Articulate communication**. E-mail enables students to contact experts and classmates on their Intranet. Furthermore, the writing an email compels them to state their needs in a concise and articulate manner.
(4) **Presentation features**. The digital format of information on the Internet facilitates its use in presentations. Text and images can be displayed in documents or slides, while video and voice can be deployed in HTML-based formats and hypertext.
(5) **Curiosity trigger**. When contents are available on the Internet, it is no longer necessary to limit the search to the course material. Teachers can therefore simply introduce new topics or different points of view within a subject through a focused Internet search, thus stimulating the student to learn more.

2.2 Virtual Labs

One of the most interesting e-learning solutions for higher education is the electronic laboratory. E-laboratories aim to fulfill the same function as traditional laboratories: to give students the opportunity to put into practice their recently acquired knowledge and skills through unlimited and repeated use [6]. Due to e-implementation, the limits of time and space are not particularly restrictive for laboratory work. It is possible to carry out laboratory experiments in a structured or open-ended enquiry form, in which students develop manual, observation, problem-solving and interpretation skills in a similar way to researchers [7]. There are two different approaches to implementing an e-lab: virtual and remote laboratories.

In a remote laboratory, students can access the equipment of a physical laboratory through a web browser. Using a remote connection to real training environments, students acquire practical skills without damage to equipment or to themselves.

Virtual laboratories, however, have no physical point of reference: students use a simulator that reproduces a real situation or implements a CAD tool (e.g. a software engineering tool, a software/hardware co-design tool, etc.). Virtual laboratories can be accessed in a straightforward manner through the use of applets embedded in a HTML page. Applets are small programs that can be used to implement simulators and other features required in a virtual lab. A considerable amount of applet repositories are freely available, and it is possible to build a virtual laboratory using these free resources.

2.3 Intelligent Tutoring Systems and Personal Assistants

In the early stages of so-called computed-based instruction (CBI), psychologists and educators rapidly became aware of the enormous potential of computers in human learning. At the end of the sixties, a group of CBI researchers began to explore a CBI approach to human cognition and learning anchored in artificial intelligence, which resulted in what are today known as intelligent tutoring systems (ITSs) [8][9][10][11].

ITS researchers adopted the idea that computers could be brought to understand students and knowledge domains, and to infer the most appropriate teaching strategies from student interaction. However, several factors meant ITSs failed and the original difficulties still remain, such as the problem of adapting the environment to the user's needs [12][13].

One of the current approaches in the ITS and related AI fields such as software agents [14] and multi-agent systems [15] is the design of personal assistants that supervise users' actions in a computer environment in order to provide assistance [16][17]. More specifically, personal assistants are interface agents that cooperate with the user in reaching an objective [18][19]. This closer cooperation between the user and the agent [20] [21][22] considerably increases user performance.

3 A Web-Based Virtual Neural Network Lab

Unlike previous Java or C++ NN labs [23][24][25], our virtual NN lab has been developed in order to provide all the elements necessary to complete a neural network project based on multi-layer perceptrons and several unsupervised NNs. Users perform a set of neural experiments under the supervision of an assistant that guides them through the various steps of a neural project, notifying them when necessary. Students log on to a portal (http://server:port/index.html) through a web browser before starting their neural experiments (Fig. 1). Additionally, there is a portal for teachers (http://server:port/professor.html) that provides administrative services (e.g. student management) and other teaching facilities, such as an editing assistant tool and an assessment module. The virtual NN lab is a web-based application written in Java that uses an object-oriented (O-O) methodology [26] and which includes the following elements:

(1) A Java server that manages all the virtual lab's services.
(2) A database engine, which stores internal information and information for an assessment of the student's progress.
(3) Java applets and other downloadable elements (Flash movies, etc.) that are automatically provided through the Web browser when necessary.

An off-line version through which users locally execute a light version of the web application is also provided, and an additional communication module can exchange information with the server when it is necessary to send assessment information. All the elements included (Java server, etc.) are freeware and are available upon request. Due to the O-O architecture, new features can easily be added to the existing virtual NN lab, as well as other neural networks not included in this version.

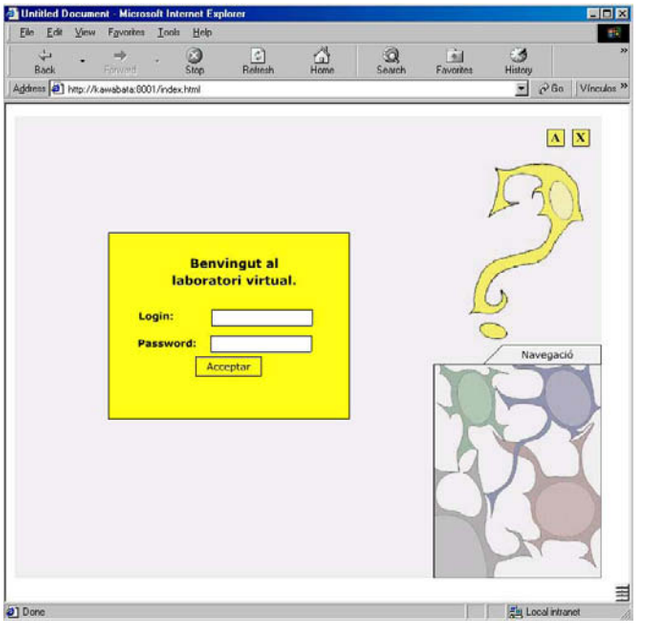


Fig. 1. The virtual NN lab's student portal

As previously mentioned, assistants designed by teachers provide students with a guide to completing several neural experiments. If mistakes are detected by the assistant in the course of the experiments, which students are free to perform in the order that they choose, a warning message will be displayed. Otherwise, assistants play a monitoring role by storing all the students' actions and reporting them to the database engine for assessment purposes. The steps that users should follow in the lab are outlined below (Fig. 2):

(1) **Target application**. Users must select a problem for the NN to solve. Currently, there are two kinds of problems or applications: classification and vector quantization.
(2) **Database formation**. An uploaded dataset or a set of graphically created samples must be provided and partitioned into training, validation and test sets.
(3) **Prototype selection and configuration**. A neural network architecture that fits the problem must be selected and its parameters must be determined.
(4) **Training**. The configured prototype is trained according to several parameters selected by the user.
(5) **Test**. Finally, generalization errors computed with an independent set are provided, as well as training and validation curves.

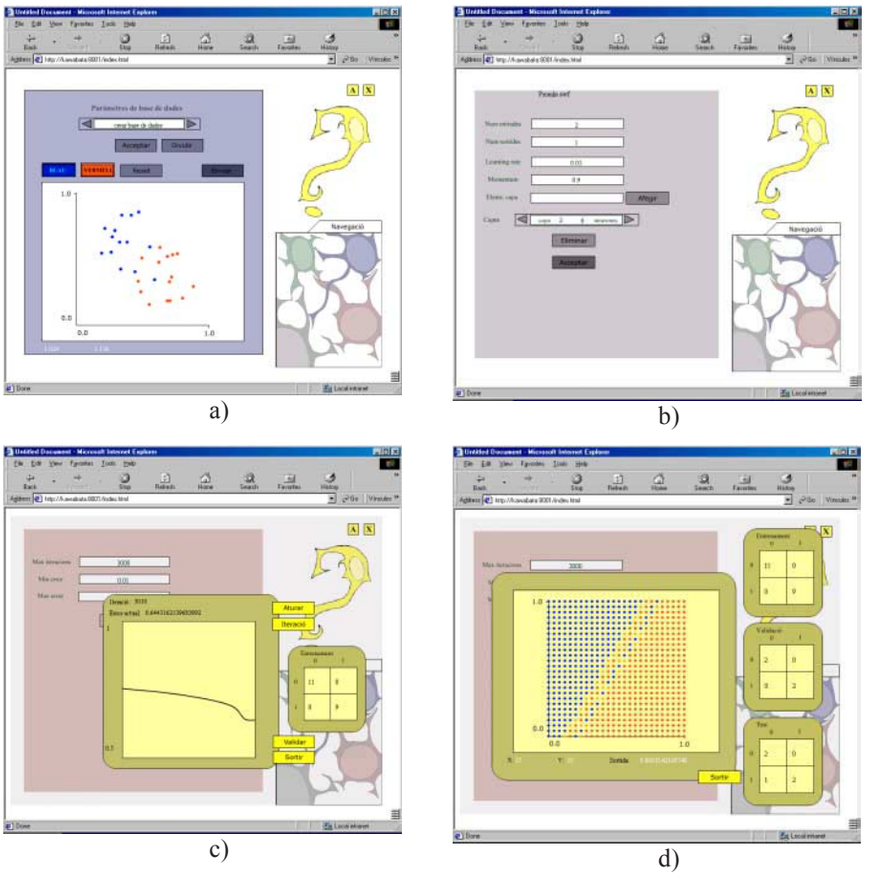


Fig. 2. Several steps in the development of a neural network project supported by the virtual lab: a) database formation, b) prototype selection and configuration, c) training, and d) test

4 Conclusions and Further Work

A web-based Java application of a virtual NN laboratory has been introduced. Our e-learning platform is based on personal assistants that support students active learning of practical skills and provide features for their assessment. Further work will focus on providing a multi-idiom version and improving the assistants intelligence.

References

[1] Juul-Olsen, K. (2001). Transforming the Current Learning Model, E-Learning Summit, Brussels 10–11 May 2001.

[2] Rosenberg, M. J. (2000). E-Learning: Strategies for Delivering Knowledge in the Digital Age. New York: McGraw-Hill Professional Publishing.

[3] Cisco Systems (2001). Learning on Demand: A Strategic Perspective. White paper. Available in digital format at www.cisco.com.

[4] CEDEFOP (2001). Non-Formal Learning: An Executive Summary. Available in digital format at http://www.trainingvillage.gr/etv/nonformal/ex_sum_EN.asp.

[5] Hartley, D. E. (2000). On-Demand Learning: Training in the New Millennium. Boston, MA: HRD Press.

[6] Network Architectures For E-Learning Applications, Revision 1.2. White Paper. Technical Education Consultants Internet Learning Solutions Group, Cisco Systems, December 2000.

[7] Brown G., Bull J., Pendlebury M. (1997). Assessing Student Learning in Higher Education. London: Routledge, chapter 7.

[8] Sleeman, D., & Brown, S. (Eds.). (1982). Intelligent Tutoring Systems. Computers and People Series. London: Academic Press.

[9] Woolf, B. (1987). Theoretical Frontiers in Building a Machine Tutor. In Kearsley, G. P. (Ed.) Artificial Intelligence and Instruction-Application and Methods, Reading, MA: Addison-Wesley, pp. 229-267.

[10] Burns, H. L., Parlett, J. W. & Redfield, and C. L. (Eds.). (1991). Intelligent Tutoring Systems: Evolutions in Design. Hillsdale, NJ: Lawrence Erlbaum Associates.

[11] Costa, E. (Ed.). (1992). New Directions for Intelligent Tutoring Systems, NATO ASI Series F: Computer and Systems Sciences, Special Program: Advanced Educational Technology, Vol. 91. Belin: Springer-Verlag.

[12] Gauthier, G. , Frasson, C. & VanLehn, K. (Eds.). (2000). Intelligent Tutoring Systems, Lecture Notes in Computer Science, Vol. 1839, Berlin: Springer Verlag.

[13] Jerinic, L. & Devedzic, V. (2000). The Friendly Intelligent Tutoring Environment: Teacher's Approach, SIGCHI Bulletin, Vol. 32 No. 1. p. 83.

[14] Bradshaw, J. M. (Ed.) (1997). Software agents, Cambridge, MA: MIT Press.

[15] Weiss, G. (Ed.) (1999). Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. Cambridge, MA: MIT Press.

[16] Edwards, P., Bayer, D., Green, C. L., Payne, T. (2001). Experience with Learning Agents which Manage Internet-Based Information. Technical Report.

[17] Perraju, T. S. (1997). Personal Assistant System for UNIX Users: A Multi Agent System Approach. Wayne State University, Department of Computer Science. Available online at http://citeseer.nj.nec.com/241126.html

[18] Lieberman, H. (2001). Autonomous Interface Agents, Media Laboratory, MIT. Available in electronic format at http://www.media.mit.edu/.

[19] Lieberman, H. (1995). Letizia: An Agent that Assists Web Browsing. Proceedings of the 1995 International Joint Conference on Artificial Intelligence, Montreal, Canada, August 1995. Available in electronic format at http://agents.www.media.mit.edu/groups/agents/publications/.

[20] Davies, J.R., Gertner, A. S., Lesh, N., Rich, C., Rickel, J., Sidner, C. L. (2001). Incorporating Tutorial Strategies into an Intelligent Assistant. Proceedings of Intelligent User Interfaces 2001, January 2001, Santa Fe, New Mexico, USA. Available in electronic format at http://www.merl.com/projects/collagen/.

[21] Rich, C., Sidner, C. L. (2001). COLLAGEN: Applying Collaborative Discourse Theory to Human-Computer Interaction. To appear in AI Magazine (Special Issue on Intelligent User Interfaces), 2001. http://www.merl.com/projects/collagen/.

[22] Rich, C. & Sidner, C. L. (1997). COLLAGEN: When Agents Collaborate with People. Proceedings of the First Conference on Autonomous Agents, Marina del Rey, CA, February 1997. Available in electronic format at http://www.merl.com/projects/collagen/.

[23] Adamo, J. M. & Anguita, D. (1995). Object Oriented Design of a Simulator for Large BP Neural Networks, in Mira, J. & Sandoval, F. (Eds.). From Natural to Artificial Neural Computation. Lecture Notes in Computer Science, Vol. 930.

[24] Fuentes, L., Aldana, J. F. & Troya, J. M. (1993). An Object-Oriented Artificial Neural Network Simulation Tool, in Mira, J., Cabestany, J. & Prieto, A. (Eds.). New Trends in Neural Computation. Lecture Notes in Computer Science, Vol. 686.

[25] Gonzalez, E. J., Hamilton, A. F., Moreno, L., Sigut, J. F. & Marichal, R. L. (2001). Evenet 2000: Designing and Training Arbitrary Neural Networks in Java, in Mira, J. & Prieto, A. (Eds.). Bio-Inspired Applications of Connectionism. Lecture Notes in Computer Science, Vol. 2085.

[26] Coleman, D. et al. (1994). Object-Oriented Development: The Fusion Method. NY: Prentice Hall.

A A o dg B d t m fo D t d t o t t o po t t o q fo g D o

ro Sa o M a o R ar o o r ra

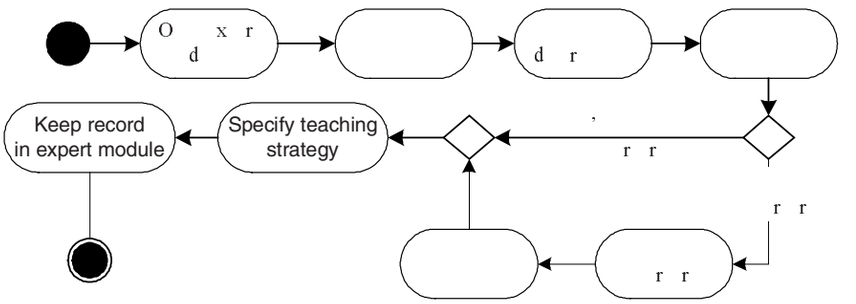
rc nd duc on nfor c D r n
nfor c n n r n nd Co u r c nc D r n
n r d d d Conc c n, C
{ c do, rcon r r, nn n} ud c c

a T a r r a ra r o o M STR a o ba y or a a o a or ora ra or q or a ra a r b o a o o r a a or or a o a o or r o r o r a o r o

u i

WWW o ay o ar or a y o o r o o a o a WWW a a o o o a ar a a r r o ' a o a y a oo ar ar o o WWW a o a y o o o r T or Sy TS o o T ry r ar a a a o o a o a b a o o or a Ma y y a b y by r a or] 6] ro r o o r ar o ra a o o o WWW a or a b y a o TS o or a a ab y o WWW a a o a ar a a o r o o o r a a o o y ar o a o ar y ar a ar o b To o a rob o r o a o a or oo o a a or or o o r r a ro o T oo bor a a a TS a r ro o y a b b a r by Br o y] T or ro a a a o a or ba o abo o q a o y o r a ar ra a a ab y o a o oo a ro a a o r ar y a o a a o o y ar a y r a or bo ra r a ro r]

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 726-733, 2003.



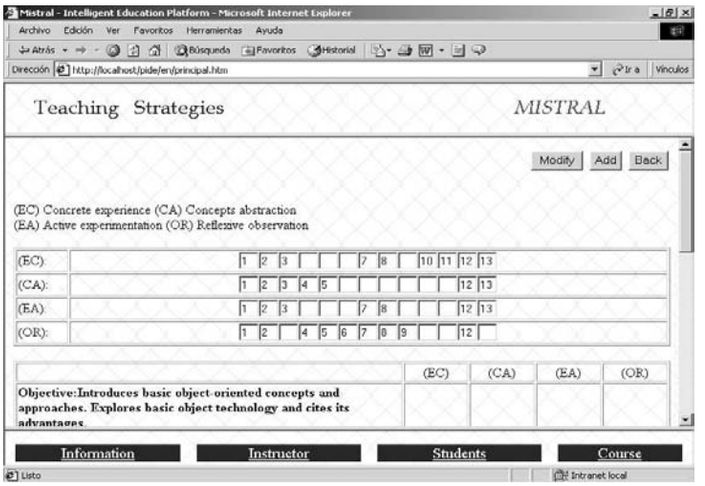
F o r o ro

a

M STR ar o a r o o o ar ro o ay T o y
r a r a a o r ro
a oo a a a a or orr rob
W a ay a M STR a o a a a a o a ra o o r
a o r r o a y a a y o a q o
o a a a ar ar o r B a ra r r
ba or o o ra y a o r r r or a o r a o o
a or r or or r or a a o T a or a
ary o or o a a o o ar r a
a ary a r q r or a ar ro
M STR ra a r a Roo r o a or a
o ro a a a o ar y a a a y a a
ra a or ar ra o o a o r a
r q r T a o o ar o a a q o
a or a ar ar ro ar y a a q r
o a o by Bay a r T o r b by
or ra a o ary r a ob r o
o a ra orr a o ro ra or
o b r or a o a a o r

M S R i

W a o o S] or a a y a o ro o a a
M or o r o M STR a ro o ra o
T o r o ro a o ra o ro



F S q o o or r y

. **n t t n p**

ra r o ro o a o r ra ro o a r r r o o a or o abo o o r o r T a a or by r r boo a r r a b o or a o abo o y ab or a q a o r ro T o abo *n r v* o r S b o *C n n* a *R v R v n* or a ob *C n n* o ob *R v R v n* or a o *v* a or o r ro a *D v* a *D r m n n n* or a a y o r o o o r a ra y q o a a o M STR o ra q o o o r o ro o ra q o a o r ar ro r a y ra ra or o b y

. **p t n p**

T o ra o ro M STR a o a o a or r or a y a a r a ay r a Roo a r or a a y ro o a o y ro o o a o oo T r a o or o ar y a o or o o a ro a o o a y a a r o q o a T a S ra o ra o a a a o ro o r o a q a ra y or a ar ar r

v

b a o r M STR ar ar ***i*** ***i*** ***chi g s*** ***gy***
o ***us*** ***di*** ***g*** a ***v u i d sig*** o r b ay
ra or o o b a

Ta STR T M r
T S S ra y

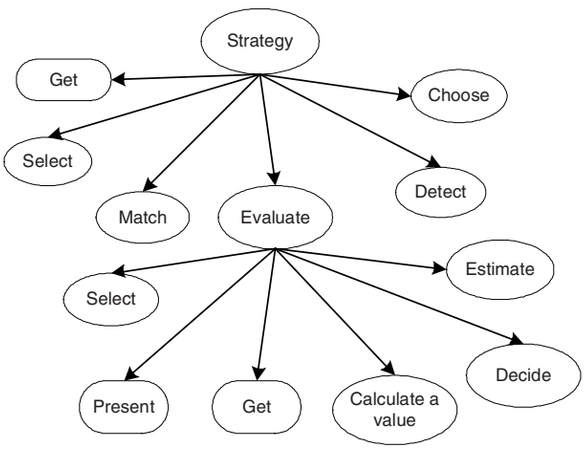
“ r or ro a ra y o a ar
ro or
R S
T r a o or a o
T T T a ra y q o a q o q o a
a o o a
S “**g t** a o **l t** ro ro ab o
a ***ch*** o r o ab a ra y o a o a o
o o ***v u*** a a ra y a ***d*** ***i***
o o o r a a y ***ch*** ***s*** a o o arry o
r ar o b
T S S ra y

A
AL A b a “ *v* a a ra y q o
a r o a o b a o r ro **MA** r
a o r ro **M** r M STR Bay
q o a oa r o o o o a r
b a a o ab a a a
r or a a o r o r r a or a o a o ro T
b a M STR a o ro ra or a r b
o o

n bt k

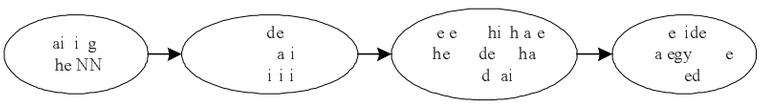
An l

r o a o a ro]]]] a a a y
a a o o or B ra o a ar ab by a o a
a a r r y o r r ro y o o
r Sa í] a a y ro o r
a r ar o o o o o ara r a ab a
a o r or a a or a ar o a o a a y
o o r o r o r T a y ab b r o o r
o o ro a o ob or T a r a a r or a
or o r y]]



g. S ra y a a a a b a

M STR o b y o a o a a ra y o
ro o o b y o a a b a or a o r



F or for d c n un ucc fu d r for ud n

M STR r by a or o r Tra ro o
ar ba o a r o a So ro o a a ra
a r a o a o a ab o o b o r a bo
o a a o ra a r a y T ra a b ra a a a r q r
a or a or o o a r a y a a ab y o a oo o
o a q o
a or a M STR a ra ar a a ar
a o o ba oo a ro o a M S ar
o ro a y a r a r o r T o r ar ar ra y
o r o o y o r o a o r o o T ra
o r o o r a boo r a a y ar r
o r or a o a q o o a or o oo a o b
ar o q o or a aba M STR o q o ar
o b y o a a or a oo or or a o a q o
a ra o r or r a a a ar r
r M o ra o r q r r
o a o o arry o a r robab y o r

l pm nt n l n t ppl t n

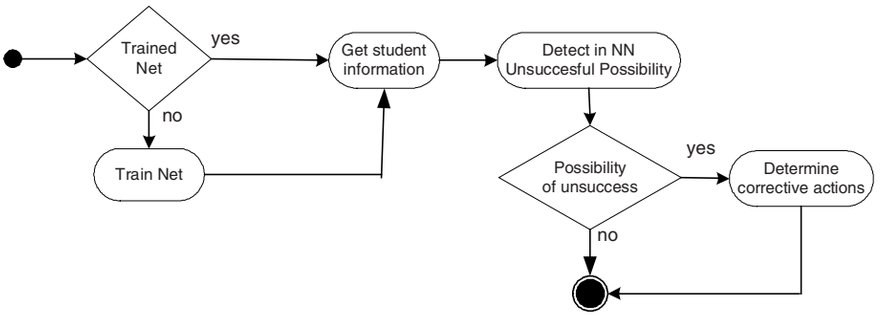
T a o ay r a a ar ab b r o o o q a y
o a or o a a o T r y a a ab a a o r
or ra or a a o a or T a a o o
o y rbo a b a o r ro a r a
a o o B a o q or o r ar
a or ra o r or o
r ra ra ar o a y o ro
a o abo o ara r Tra
o a b r o r a or a q a ra rror o r
a a o o o r ro r o o a o
o r ar
a a M STR ara r o ay r a a a a a ab
M STR a b o o a o r r a y S a a a b ar or
r o ra ro a o a a b a b ar
a o r o a r a or a r a ar ab o a r o
o a ar ’ a o a ar ab a r o or ba
oo r y ra b or r y or r o o r
r a ar ab r o a o r a or ar ar a o a
or a a o S r a ar ra y S o a o r
a a o o r a a o or a So a Wor o a y
Mo r or o

n ng n t t t

To a a a a q o rob ra a a ab or a o
b o o r y o o o r a r o
y ay M STR a r ro r a ra
ar a a ro a or ar o b a a a or ra
b by r
ar r o y o ar ab a o a a o o
r or a o r
o r r a a b ra or o ro a o o o
a a or a o r a a a a r r T *l n* o
a r a or ar o o y o
orr o a a or a o o b r o a r r
ro r o o o a

An An l n b t nt gn t M AL

o r o M STR “ a y a b
r or a r r y ar o r T o r a
roo or r a a o b M STR a a r
o o T o r r r a a o ar ar ’ ara r a
r o o ro ra a a a o oo r a



g. or “ b a

o o r a ab y or oo a a o o r or o b a ro ra a a a oo o oo Ra o a Ro ro a o or a a a a q o o T o r r r a a o ar y or a M STR a a a y a a a or a q a ra y or a o o a o q o a by M STR o r r r a a o o o a by a o a r o r a a o M STR ro o a r o a r ro or r a a o o a ar ro T ro o a r or a a ar r a a or a o o r a ab y or a r a r a ab a r T r a abo o o r r ar y a ay a r y ar ro T o b y o or r r or o o q a o bo r or a o a a a r a

u i

o M STR ro Bay a ra or q a o a r o a a o o r r y o o a o a a o by ra a ra y o ’ r q r

or a r a oo a a o o a a y o r r r a o r o T ar ar a o a o a b M STR a y oa a a b a ro o o o r a or a a o r

ro or a a o o o a or o ’ y r ra a o r o b or a a o ro a o a r a ra M STR or o ary ro o o or a o o a ar a ro a or o ar ar ’ ara r M STR a ary o o a

a o a ar r a ary q o a o r or
a r q r
or a b o a by or ora ra q o
M STR ro robab y or a b a o b o
ro o orr a o o a ar ro a r a o

R

Br o y ob a a a a a y r a
r a or r r a b r
Too r a a o a ro r oo or r a o
oo o o r
r a a S Torr T a or q
a r a a Ma a r a o
a a r a ar *r L n n v r*
n m n r v r r n z n r n m n
r mn n ñ n ñ nz m v
r n r a o
a 6 *n m n rn n m L r r*
o ro
6 TS ro o r a o a o r “ T or
Sy Mo r a a a a Mo r a a a a Mo r a a a a
6 T a S Mo r a a a a ra
o b a *Inv n r r n z* a W r R r
r y
a r S y o r *D m n m r n* o
a a o
Sa í *D n mn r m n n*
n r r m n r n r n Wor r or a
So o a a o o o ía a a M
Sa í *In n r n m n r n m n*
r n rn n : *r n v* Wor
r or a a o a y r ar a a M
S r b r r a r oo R S a bo a
W W W a B *n n n r n n m n m n*
C mm n S M T r a br Ma
S a a Bro S T or Sy or
a r

A recurrent neural network model for the p-hub problem

E. Domínguez[1], J. Muñoz[1], and E. Mérida[2]

[1] Department of Computer Science
[2] Department of Applied Mathematics
E.T.S.I.Informatica, University of Malaga
Campus Teatinos s/n, 29071 – Malaga, Spain
{enriqued, munozp}@lcc.uma.es, merida@ctima.uma.es

Abstract. The p-hub problem is a facility location problem that can be viewed as a type of network design problem. Each node, within a given set of node, sends and receives some type of traffic to and from the other nodes. The hub location must be chosen from among these nodes to act as switching points for the traffic. Hubs are facilities that serve as transshipment and switching points for transportation and telecommunication systems with many origins and destinations. In this paper we consider the uncapacitated, single allocation, p-hub median problem. In the single allocation, each nonhub node must be allocated to exactly one of the p hubs. We provide a reduced size formulation and a competitive recurrent neural model for this problem. The neural network consists of a two layers (allocation layer and location layer) of np binary neurons, where n is the number of nodes and p is the number of hubs. The process units (neurons) are formed in groups, where one neuron per group is active at the same time and neurons of different groups are updated in parallel. Computational experience with another neural networks is provided using the data given in the literature.

1 Introduction

The problem of locating hub facilities arises frequently in the design of telecommunication networks, as well as in airline passenger flow. Discrete hub location problems involve locating a set of fully interconnected facilities called hubs, which serve as transshipment and switching points for traffic between specified origins and destinations. A non-negative flow is associated with every origin-destination pair and an attribute such as distance, time or cost is associated with movement. The term of cost is used throughout this paper to represent the attribute of interest. In hub systems, origin-destination movements are generally via one or two hubs. As long as the cost of movement is a nondecreasing function of distance, no origin-destination movements are via more than two hubs, since hubs are fully interconnected.

The problem addressed in this paper models the situation where there are n nodes in a network, and p of these nodes are to be designed as hubs. Each node

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 734-741, 2003.

in the network can interact with each other only via the hubs to which they have been allocated. More specifically, the problem dealt with in this paper is the uncapacitated, single allocation, p-hub median problem (USApHMP).

A quadratic integer programming formulation for USApHMP was proposed by O'Kelly [8]. Since O'Kelly original formulation, several researchers have used various heuristics to solve this problem. The USApHMP is known to be NP-complete, like related p-median and p-center problems. While these other problems become trivial once the locations are known however, the USApHMP remains NP-complete even if the hub location are known (see [9]). This is due to the lack of a simple rule for solving the allocation phase of the USApHMP.

O'Kelly [8] considered the use of two heuristics for solving a USApHMP which models flights paths of an airline company, and attempts to assign each airport to a fixed hub. Thus, the problem was reduced from location-allocation problem to a problem alone. Klincewicz [7] examined ways of avoiding convergence of such heuristics to sub-optimal local minima by using tabu search and GRASP strategies, although the problem being considered was still the simplified problem of allocating nodes to a fixed hubs using a minimum distance rule. Skorin-Kapov and Skorin-Kapov [10] considered the use of tabu search for solving the complete location-allocation problem. Aykin [1] devised various others heuristics, Ernst and Krishnamoorthy [5] applied a simulated annealing heuristic and Smith, Krishnamoorthy and Palaniswami [6] considered a modified Hopfield netowrk to solve the USApHMP. In this paper we proposed a recurrent neural model for solving the USApHMP that we applied usefully to related problem like the p-median problem [3].

The paper is organized as follows. In section 2 we review the problem formulations and we propose a new reduced size formulation. Section 3 presents the proposed recurrent neural model. Illustrative simulations and computational results with the well known 1970 Civil Aeronautics Board (CAB) data set are described in section 4. Finally, section 5 provides a summary and conclusions.

2 Problem Formulation

The hub location problem we consider here can be described as follows: we are given the location of a set of nodes $N = \{1, 2, ...n\}$, the volume of flow (w_{ij}) that must be shipped between each origin-destination pair and the cost per unit flow (c_{ij}) between each origin-destination pair. From the set N of nodes we select p of them to be hubs. Hubs are consolidation or switching points for flow and they are fully connected. All flow travels via hubs and each non-hub node must be allocated to a unique hub node. The location of the hubs and the allocation of the nodes is chosen so that the total cost of the system is minimized. It should be noted that all flow that must be shipped between nodes, has three separate components: collection (origin node to hub), transfer (hub to hub) and distribution (hub to destination node).

O'Kelly [8] first formulated the USApHMP as a quadratic integer program. This formulation has $O(n^2)$ variables, even so this problem is difficult to solve

due to the non-convexity of the objective function. Subsequently a mixed integer linear program with $O(n^4)$ variables was developed by Campbell [2] to obviate the non-convexity of the objective function. Ernst and Krishamoorthy [5] developed a reduced mixed integer linear program using $O(n^3)$, and recently, Ebery [4] presented a formulation with $O(n^2)$ variables. In this paper, we proposed a new reduced formulation for the USApHMP using $O(n)$ variables. The proposed formulation is defined as follow

Minimize

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{q=1}^{p}\sum_{k=1}^{n}\left(\beta w_{ik}c_{ij}+\gamma w_{ki}c_{ji}+\alpha\sum_{m=1}^{n}\sum_{r=1}^{p}w_{ik}c_{jm}x_{kr}y_{mr}\right)x_{iq}y_{jq} \tag{1}$$

Subject to

$$\sum_{q=1}^{p}x_{iq}=1 \quad i=1,..N \tag{2}$$

$$\sum_{j=1}^{N}y_{jq}=1 \quad q=1,..p \tag{3}$$

where

$x_{iq}=\begin{cases}1 \text{ if the node } i \text{ is allocated to the cluster } q\\ 0 \text{ otherwise}\end{cases}$

$y_{jq}=\begin{cases}1 \text{ if the node } j \text{ is the hub in the cluster } q\\ 0 \text{ otherwise}\end{cases}$

w_{ij} is the amount of flow from the node i to the node j

c_{ij} is the transportation cost associated between the nodes i and j

α is the transfer coefficient

β is the collection coefficient

γ is the distribution coefficient

In (1) the first two terms inside the brackets evaluate the cost of assigning a node to its hub for outgoing and incoming flows respectively. These terms are multiplied by two parameters respectively: β (collection coefficient) and γ (distribution coefficient). The third component (the double summation over m and r) counts the costs of those interactions, which must flow between hubs. These inter-hub costs are multiplied by a parameter α to reflect the scale effects in inter-facility flows. Constraint (2) ensures that each node is allocated to a unique cluster and restriction (3) ensures that one and only one hub is opened in each cluster.

3 Competitive Recurrent Neural Model

The proposed neural network consists of two layers (allocation layer and location layer) of interconnected binary neurons or processing elements. Each neuron i has an input h_i and an output S_i. In order to design a suitable neural network for this problem, the key step is to construct an appropriate energy function E for

which the global minimum is simultaneously a solution of the above formulation. The simplest approach to constructing a desired energy function is the penalty function method. The basic idea in this approach is to transform the constrained problem into an unconstrained one by adding penalty function terms to the objective function (1). These terms cause a high cost if any constraint is violated. More precisely, some or all constraints are eliminated by increasing the objective function by a quantity which depends on the amount by which the constraints are violated. That is, the energy function of the neural network is given by the Liapunov energy function defined as

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{q=1}^{p}\sum_{k=1}^{n}\left(\beta w_{ik}c_{ij}+\gamma w_{ki}c_{ji}+\alpha\sum_{m=1}^{n}\sum_{r=1}^{p}w_{ik}c_{jm}x_{kr}y_{mr}\right)x_{iq}y_{jq}+ \\ +\lambda_1\sum_{i=1}^{n}\left(1-\sum_{q=1}^{p}x_{iq}\right)+\lambda_2\sum_{q=1}^{p}\left(1-\sum_{j=1}^{n}y_{jq}\right) \tag{4}$$

where $\lambda_i > 0$ are penalty parameters that they determine the relative weight of the constraints. The penalty parameters tuning is an important problem associated with this approach. In order to guarantee a valid solution and avoid the parameter tuning problem, we will divide our neural network in disjoint groups according to the two restrictions, that is, for the USApHMP with n points, we will have n groups, according to restriction 2, plus p groups, according to restriction 3. Then, we will reorganize our neurons in two matrices (one matrix per neuron type) where a group is represented by a row or column of the matrix according to neuron type.

$$\begin{array}{ll} & \begin{array}{cccc} y_{11} & y_{12} & \cdots & y_{1p} \\ y_{21} & y_{22} & \cdots & y_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n1} & y_{n2} & \cdots & y_{Np} \\ \downarrow & \downarrow & & \downarrow \\ \text{group n}+1 & \text{group n}+2 & & \text{group n}+\text{p} \end{array} \\ \begin{array}{l} x_{11}\ x_{12}\cdots x_{1p}\rightarrow \text{group 1} \\ x_{21}\ x_{22}\cdots x_{2p}\rightarrow \text{group 2} \\ \vdots\quad\vdots\quad\ddots\quad\vdots\qquad\quad\vdots \\ x_{n1}\ x_{n2}\cdots x_{np}\rightarrow \text{group n} \end{array} & \end{array}$$

Fig. 1. Neuron organization of the neural network for the USApHMP with n points.

Figure 1 shows two matrices, the first matrix contains the allocation neurons and the second contains the location neurons. The allocation neurons inside same group are in the same row of the matrix, and the location neurons inside same group are in the same column.

In this model one and only one neuron per group must have one as its outputs, so the penalty terms are eliminated from the objective function. The neurons inside same group are updated in parallel. Then we should ought introduce the notion of group update. Observe that the groups are updated sequentially. Then, the energy function of the neural network is reduced to

$$E = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{q=1}^{p}\sum_{k=1}^{n}\left(\beta w_{ik}c_{ij} + \gamma w_{ki}c_{ji} + \alpha\sum_{m=1}^{n}\sum_{r=1}^{p} w_{ik}c_{jm}x_{kr}y_{mr}\right)x_{iq}y_{jq} \quad (5)$$

We avoid the parameter tuning problem of λ_1 and λ_2 in the expression (4) with the new model due to the energy function of the new model (5) do not have penalty terms.

The inputs of each neuron of the network are

$$h_{x_{iq}} = -\sum_{j=1}^{n}\sum_{k=1}^{n}\left(\beta w_{ik}c_{ij} + \gamma w_{ki}c_{ji} + \alpha\sum_{m=1}^{n}\sum_{r=1}^{p} w_{ik}c_{jm}x_{kr}y_{mr}\right)y_{jq} \quad (6)$$

$$h_{y_{jq}} = -\sum_{i=1}^{n}\sum_{k=1}^{n}\left(\beta w_{ik}c_{ij} + \gamma w_{ki}c_{ji} + \alpha\sum_{m=1}^{n}\sum_{r=1}^{p} w_{ik}c_{jm}x_{kr}y_{mr}\right)x_{iq} \quad (7)$$

where $h_{x_{iq}}$ is the activation potential of allocation neuron iq and $h_{y_{jq}}$ is the activation potential of the location neuron jq.

The central property of the proposed network is that the computational energy function always decrease (or remains constant) as the system evolve according to its dynamical rule

$$x_{iq}(k+1) = \begin{cases} 1 \text{ if } h_{x_{iq}}(k) = \max\limits_{1\le j\le n}\{h_{x_{jq}}(k)\} \\ 0 \text{ otherwise} \end{cases} \quad (8)$$

$$y_{jq}(k+1) = \begin{cases} 1 \text{ if } h_{y_{jq}}(k) = \max\limits_{1\le i\le n}\{h_{y_{iq}}(k)\} \\ 0 \text{ otherwise} \end{cases} \quad (9)$$

Theorem 1. *Let M be a binary neural network characterized by a Liapunov energy function (5) where the inputs of the neurons are computed by (6) and (7). If only one group g is selected for updating at every time k and the dynamics of the neural network are given by expressions (8) and (9), then the energy function is guaranteed to decrease.*

Proof. If we consider discrete time dynamics then the energy difference for any change of the state of any neurons of the network is

$$\Delta E(k) = E(k+1) - E(k) =$$

$$= \sum_{i=1}^{n}\sum_{q=1}^{p} h_{x_{iq}}(k)x_{iq}(k) - \sum_{i=1}^{n}\sum_{q=1}^{p} h_{x_{iq}}(k+1)x_{iq}(k+1) =$$

$$= \sum_{j=1}^{n}\sum_{q=1}^{p} h_{y_{jq}}(k)y_{jq}(k) - \sum_{j=1}^{n}\sum_{q=1}^{p} h_{y_{jq}}(k+1)y_{jq}(k+1)$$

We introduce now the notion of group update, that is, instead of selecting a single neuron for update we can select a group g containing a number of m neurons. Then, the difference in the energy that would result if only the states of the neurons in the group g was altered depends on the type of group. If all neurons contained in the selected group ($g = i$) are allocation neurons (x_{iq}), namely allocation group, the difference in the energy is

$$\Delta E_g(k) = \sum_{q=1}^{p} h_{x_{iq}}(k)x_{iq}(k) - \sum_{q=1}^{p} h_{x_{iq}}(k+1)x_{iq}(k+1)$$

If all neurons contained in the selected group ($g = n+q$) are location neurons (T_{jq}), namely location group, the difference in the energy is

$$\Delta E_g(k) = \sum_{j=1}^{n} h_{y_{jq}}(k)y_{jq}(k) - \sum_{j=1}^{n} h_{y_{jq}}(k+1)y_{jq}(k+1)$$

Let us suppose now that at time k the allocation neuron x_{ir} is the only one that is "on" in group i and that allocation neuron x_{io} is the candidate neuron in group i that is going to be "on" at time $k+1$. Then we have that

$$x_{iq}(k) = \begin{cases} 1 \text{ if } q = r \\ 0 \text{ otherwise} \end{cases} \quad x_{iq}(k+1) = \begin{cases} 1 \text{ if } q = o \\ 0 \text{ otherwise} \end{cases}$$

By substituting this values we have that

$$\Delta E_i(k) = h_{x_{ir}}(k) - h_{x_{io}}(k+1) = h_{x_{ir}}(k) - h_{x_{io}}(k)$$

Hence, we have that if the neuron with the maximum input $h_{x_{io}}$ per group i is always selected as the candidate neuron x_{io}, then the energy descent is guaranteed. Thus, the value of the energy decrease $\Delta E_i(k)$ is the maximum possible at every time k.

Let us suppose now that at time k the location neuron y_{rq} is the only one that is "on" in group $n+q$ and that location neuron y_{oq} is the candidate neuron in group $n+q$ that is going to be "on" at time $k+1$. Then we have that

$$y_{jq}(k) = \begin{cases} 1 \text{ if } j = r \\ 0 \text{ otherwise} \end{cases} \quad y_{jq}(k+1) = \begin{cases} 1 \text{ if } j = o \\ 0 \text{ otherwise} \end{cases}$$

By substituting this values we have that

$$\Delta E_{n+q}(k) = h_{y_{rq}}(k) - h_{y_{oq}}(k+1) = h_{y_{rq}}(k) - h_{y_{oq}}(k)$$

Hence, we have that if the neuron with the maximum input $h_{y_{oq}}$ per group $n+q$ is always selected as the candidate neuron y_{oq}, then the energy descent is guaranteed. Thus, the value of the energy decrease $\Delta E_{N+q}(k)$ is the maximum possible at every time k. □

The following procedure describes the neural network algorithm (NNA).

1. Set the initial state by randomly setting the output of one neuron in each of $n + p$ groups to be one and all the others neurons in the group to be zero.
2. Evaluate the initial value of the energy function by expression (5).
3. Select a group g where $1 \leq g \leq n + p$.
4. Compute the inputs of the neurons in the group g, by expression (6) if $1 \leq g \leq n$ or by expression (7) otherwise.
5. Update neurons by expression (8) if $1 \leq g \leq n$, or update neurons by expression (9) if $n + 1 \leq g \leq n + p$.
6. Repeat from step 3 until no more changes.

On step 3 we select a group randomly or sequencely. On step 5, if there are different neurons with the maximum input value, the algorithm must randomly select one of them.

4 Simulation Results

The data for this study are based on the well known CAB (Civil Aeronautics Board) data sets from the literature (see [8, 6]). Problems of size $n = 10$ and 15 are extracted from this data set, while further problems are generated by varying the number of hubs $p \in \{2, 3, 4\}$ and the transfer cost $\alpha \in \{0.2, 0.4, 0.6, 0.8, 1\}$. The values of the collection and distribution coefficients are fixed at $\beta = \gamma = 1$. Exact results are again provided using the linear programming approach of Ernst and Krishnamoorthy [5]. The results are compared to the Hopfield network (HN) and the modified hill-climbing Hopfield network (HCHN) provided by Smith et al. [6]. All algorithms were coded in the Matlab programming language and run on an Origin 2000 (Silicon Graphics Inc.) multiprocessor operated under IRIX 6.5 with 16 CPUs MIPS R1000.

The results presented in Table 1 demonstrate quite clearly that the proposed NNA are able to compete effectively with the Hopfield neural networks approaches proposed by Smith et al. [6] in finding optimal or near-optimal solutions to the CAB data sets. The HN often converges to a poor quality solution, since it becomes caught in the first local minima it encounters. Although the HCHN considerably improves the quality of solutions, with optimal solutions being located in 73% of the CAB problem instances. However, the amount of CPU time required to simulated the proposed NNA is the main advantage with respect the Hopfield network approaches. Thus, the CPU time for the HN and HCHN simulations are several orders of magnitude greater than those for the NNA. Moreover, the memory requirements for these simulations (HN and HCHN) also make it difficult to obtain solutions for problems greater than $n = 20$.

5 Conclusions

In this paper, we have reviewed the uncapacitated single allocation p-hub problem. We have reduced the number of variables and constraints of the formulations provided by several authors [8, 5, 2]. As another neural solution approach,

Table 1. Results of CAB data sets for NNA, HN, HCHN

n	p	α	Optimal	NNA	HN	HCHN	n	p	α	Optimal	NNA	HN	HCHN
10	2	0,2	615,99	0,08%	0,00%	0,00%	15	2	0,2	981,28	0,16%	0,00%	0,00%
		0,4	674,31	0,05%	0,00%	0,00%			0,4	1062,63	0,14%	0,00%	0,00%
		0,6	732,63	0,03%	0,00%	0,00%			0,6	1143,97	0,09%	0,00%	0,00%
		0,8	790,94	0,02%	0,00%	0,00%			0,8	1190,77	0,08%	3,00%	0,20%
		1	835,81	0,02%	6,60%	1,60%			1	1221,92	0,07%	7,20%	0,50%
	3	0,2	491,93	0,13%	18,80%	0,00%		3	0,2	799,97	0,09%	0,20%	0,00%
		0,4	567,91	0,08%	15,90%	0,00%			0,4	905,1	0,08%	2,10%	0,00%
		0,6	643,89	0,08%	1,70%	0,00%			0,6	1009,93	0,16%	0,60%	0,00%
		0,8	716,98	0,07%	0,00%	0,00%			0,8	1099,51	0,12%	3,50%	0,00%
		1	776,68	0,05%	1,00%	0,40%			1	1168,68	0,13%	1,40%	1,00%
	4	0,2	395,13	0,24%	0,00%	0,00%		4	0,2	639,78	0,11%	0,00%	0,00%
		0,4	493,79	0,10%	1,10%	0,00%			0,4	779,71	0,11%	4,60%	0,00%
		0,6	577,83	0,10%	4,80%	0,00%			0,6	910,21	0,14%	2,80%	0,00%
		0,8	661,42	0,09%	6,90%	0,40%			0,8	1026,52	0,15%	1,90%	0,00%
		1	736,26	0,08%	0,70%	0,40%			1	1118,23	0,18%	3,70%	1,20%

we considered a competitive recurrent neural model. In the computational experience, we have shown that the proposed neural model worked well, only 0,1% average error for the CAB data sets, and found optimal or near-optimal solutions quickly. Therefore, the proposed recurrent neural network might be used for large problems.

References

1. T. Aykin. The hub location an routing problem. *European Journal of Operational Research*, 83:200–219, 1995.
2. J. F. Campbell. Integer programming formulations of discrete hub location problems. *European Journal of Operational Research*, 72:387–405, 1994.
3. E. Domínguez and J.Mu noz. An efficient neural network for the p-median problem. *Lecture Notes in Aritifitial Intelligence*, 2527:460–469, 2002.
4. J. Ebery. Solving large single allocation p-hub problems with two or three hubs. *European Journal of Operational Research*, 128:447–458, 2001.
5. A. Ernst and M. Krishnamoorthy. Efficient algorithms for the uncapacitated single allocation p-hub median problem. *Location Science*, 4:139–154, 1996.
6. M. Krishnamoorthy K. Smith and M. Palaniswami. Neural versus traditional approaches to the location of interacting hub facilities. *Location Science*, 4(3):155–171, 1996.
7. J. G. Klincewicz. Avoiding local minima in the p-hub location problem using tabu search and grasp. *Annals of Operational Ressearch*, 40:283–302, 1992.
8. M. E. O'Kelly. A quadratic integer program for the location of interacting hub facilities. *European Journal of Operational Research*, 32:393–404, 1987.
9. James G. Morris Robert F. Love and George O. Wesolowsky. *Facilities Location Models & Methods*. North-Holland, 1998.
10. D. Skorin-Kapov and J. Skorin-Kapov. On tabu search for the location of interacting hub facilities. *European Journal of Operational Research*, 73:502–509, 1994.

A comparison of the performance of SVM and ARNI on Text Categorization with new filtering measures on an unbalanced collection *

Elías F. Combarro[2], Elena Montañés[1], José Ranilla[1], Javier Fernández[1]

[1] Artificial Intelligence Center, University of Oviedo. Campus de Viesques, Gijón (Asturias) Spain
[2] Computer Science Department, University of Oviedo. Campus de Viesques, Gijón (Asturias) Spain

Abstract. Text Categorization (TC) is the process of assigning documents to a set of previously fixed categories. A lot of research is going on with the goal of automating this time-consuming task due to the great amount of information available. Machine Learning (ML) algorithms are methods recently applied with this purpose. In this paper, we compare the performance of two of these algorithms (SVM and ARNI) on a collection with an unbalanced distribution of documents into categories. Feature reduction is previously applied with both classical measures (*information gain* and *term frequency*) and 3 new measures that we propose here for first time. We also compare their performance.

1 Introduction

One of the main tasks in processing large collections of text files is that of assigning the documents of a corpus into a set of previously fixed categories, what is known as TC. Formally, TC consists of determining whether a document d_i (from a set of documents D) belongs or not to a category c_j (from a set of categories C).

Since TC involves a great amount of features and most of them could be irrelevant or noisy [18], a previous feature reduction could improve the performance of the classifiers.

The aim of this paper is to study the performance of TC with feature reduction using Support Vector Machines (SVM) [10] and ARNI ([15]), a ML algorithm that constructs decision trees using a measure of the quality of the tests which is called *impurity level*. To tackle feature reduction, we used some measures previously applied by other authors and some new measures proposed by us, based on concepts of the ML environment.

The rest of the paper is organized as follows. In Section 2 it is discussed some previous work. In Section 3 the process of TC is described. The corpus description and the experiments are exposed in Section 4. Finally, in Section 5 some conclusions and ideas for further research are presented.

* The research reported in this paper has been supported in part under MCyT and Feder grant TIC2001-3579 and FICYT grant BP01-114.

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 742-749, 2003.

2 Previous Work

The aim of TC is to find out whether a document is relevant to a category or not. Typically, the task of assigning a category to a document from a finite set of m categories is commonly converted into m binary problems each one consisting of determining whether a document belongs to a fixed category or not. This transformation makes possible the use of a wide number of binary classifiers to solve the multi-category classification problem [2], [10] (*one-against-the-rest*). The rest of the section briefly presents some previous work about classification and feature reduction.

2.1 Classification Algorithms

In the literature about TC two main streams can be found: statistical methods and, recently, algorithms from the field of artificial intelligence.

Among the statistical methods, one of the first to be used on TC was Rocchio's algorithm [17]. Other statistical approaches are the naive Bayes classifier [10] and the instance-based algorithm called KNN [22].

In ML field we find algorithms that induce decision trees (or, more generally, classification rules) from previously labeled examples: C4.5 of Quinlan [14] used for TC in [10]; ARNI (Trees and Rules based on the Impurity Level) [15] previously used for TC in [8] or RIPPER [4] used in TC in [5].

One of the latest approaches is SVM used for TC in [10], which is a universal binary classifier able to find out threshold functions to separate the examples of a certain category from the rest.

2.2 Feature Reduction

Feature selection is one of the techniques of feature reduction which has been shown to be more adequate for TC. It is performed by keeping the words with highest score according to a predetermined measure of the importance of the word. Among the measures previously used in TC for this purpose we found the traditional Information Retrieval (IR) measures tf and $tf \times idf$ in [12], [22] and Information Theory (IT) measures like *information gain* or *mutual information* in [22].

Previous feature reduction [18] is typically carried out eliminating meaningless words like articles, conjunctions, auxiliary verbs which are called *stop words*. Additionally, the reduction of words to their common roots or stems, which is called *stemming* is usually performed. Simultaneously, there are two different ways of feature selection: local, which consists of the words occurring in documents of each category; and global, which consists of the words occurring in all the documents of all categories. In this paper local reduction is performed since it has been proved that it performs better [12].

3 Process of Text Categorization

In this section it is described how we perform the process of TC.

3.1 Document Representation

The most widely used document representation is that of *vector of words* or *bag of words* (see [18]), which consists of identifying each document with a numerical vector whose components evaluate the importance of the different words in the document. We used tf (the absolute frequency of the word in the document), which has been widely adopted in previous work [10] and [22].

3.2 Feature Reduction

Here it is described the measures for feature reduction followed in this paper.

First, we consider (after eliminating *stop words* and performing *stemming*, following Porter [13]) feature reduction by weighting over local vocabularies.

As traditional IR scoring measure we chose tf following the results presented in [12]. This measure simply counts the number of occurrences of each word.

Among the IT measures we select *information gain* since it has been showed to perform very well [22]. Its expression is

$$IG(w,c) = P(w)P(c/w)\log(\frac{P(c/w)}{P(c)}) + P(\overline{w})P(c/\overline{w})\log(\frac{P(c/\overline{w})}{P(c)})$$

where $P(w)$ is the probability that the word w appears in a document, $P(c/w)$ is the probability that a document belongs to the category c if we know that the word w appears in it, $P(\overline{w})$ is the probability that the word w does not appear in a document and $P(c/\overline{w})$ is the probability that a document belongs to the category c if we know that the word w does not occur in it. Usually, these probabilities are estimated by means of the corresponding relative frequencies.

The measures proposed in this paper are taken from the ML environment. These measures have been applied to quantify the quality of the rules induced by a ML algorithm. In order to be able to adopt these measures we propose to associate each pair (w,c) to the following rule: *If the word w appears in a document, then that document belongs to category c.* From now on, this rule is denoted by $w \to c$. In this way the quantification of the importance of a word w in a category c is reduced to the quantification of the quality of the rule $w \to c$. Some notation must be introduced in order to define the measures: for each pair (w,c), $a_{w,c}$ denotes the number of documents of the category c in which w appears, $b_{w,c}$ denotes the number of documents that contain the word w but they do not belong to the category c, $c_{w,c}$ denotes the number of documents of the category c in which w does not appear and finally $d_{w,c}$ denotes the number of documents which neither belong to category c nor contain the word w.

In general, the rule quality measures are based on the percentage of successes and failures. The Laplace measure used in CN2 [3] and in RISE [7] and the *difference* that supposes a simplification of the *accuracy* used by Muggleton [11] and by Fürnkranz & Widmer [9] are measures of this kind. The Laplace measure is obtained by means of $L(w \to c) = \frac{a_{w,c}+1}{a_{w,c}+b_{w,c}+s}$, where s is the number of categories. The particularity of this measure is that it penalizes the words that

appear in few documents. The *difference* penalizes the number of documents of the category in which the word appears by subtracting from them the number of documents of the rest of categories in which the word also appears. Its expression is $D(w \rightarrow c) = a_{w,c} - b_{w,c}$. Besides, there exist other rule quality measures that additionally take into account the number of documents in which the word occurs in the category and the unbalanced distribution of documents over the categories. A measure of this kind is the *impurity level* (IL) proposed by Ranilla et al. [16] and motivated by the criterium used in IB3 [1]: If a rule is applied n times and has m successes, the confident interval of its percentage of success is determined by the formula taken from Spiegel [20], which takes into account the percentage of success ($\frac{m}{n}$ or$\frac{a_{w,c}}{a_{w,c}+b_{w,c}}$). The distribution of examples over the categories is considered by means of the *canonic chance rule* (denoted by $\rightarrow c$, which is the rule that do not have any antecedent and has the category as consequent). Then, IL is defined by the overlapping degree between the confident interval of the rule and the confident interval of its correspondent canonic rule $IL(w \rightarrow c) = 100 \cdot \frac{CI_d(\rightarrow c) - CI_i(w \rightarrow c)}{CI_d(w \rightarrow c) - CI_i(w \rightarrow c)}$.

Since in IR is important to take into account the positive failures, a variant of each measure is proposed. Each variant consists of adding the parameter $c_{w,c}$ to the previous expressions, that is, to take into account the number of documents of the category in which w does not appear. The effect of this modification is to penalize that words which appear in few documents of the category. The new expressions are (the percentage of successes p_{ir} is only what it changes in the definition of IL) $L_{ir}(w \rightarrow c, \overline{w} \rightarrow \overline{c}) = \frac{a_{w,c}+1}{a_{w,c}+b_{w,c}+c_{w,c}+2}$, $D_{ir}(w \rightarrow c, \overline{w} \rightarrow \overline{c}) = a_{w,c} - b_{w,c} - c_{w,c}$ and $p_{ir}(w \rightarrow c, \overline{w} \rightarrow \overline{c}) = \frac{a_{w,c}}{a_{w,c}+b_{w,c}+c_{w,c}}$ respectively.

3.3 Classification

In this section, ARNI and SVM, the algorithms used in this paper, are described.

ARNI [15] is based on the principles of Hunt for the construction of decision trees. In this approach the original train set of examples is recursively divided into subsets that either only contain examples of one category or they are near of it. The division is made according to a predefined test over the features. The test adopted in ARNI is based on *the impurity level*.

SVM is a good handler of many features and it deals very well with sparse examples. These are universal binary classifiers able to find out linear or non-linear threshold functions to separate the examples of a certain category from the rest and they are based on the *Structural Minimization Risk* principle from computational learning theory [21]. We adopt this approach with a linear threshold function since the most of TC problems are linearly separable [10].

3.4 Evaluation of the Performance

Two measures of effectiveness commonly used in IR have been widely adopted [19]: *precision* and *recall*. Given the trade off existing between them [19], it is

adequate to combine them. One adequate combination is F_1 which gives the same relevance to both and it is defined as $F_1 = \frac{1}{0.5\frac{1}{Pr}+0.5\frac{1}{Rc}}$.

To compute the global performance over all the categories there exist two different approaches: *macroaveraging* (averaging the *precision*, *recall* and F_1) or *microaveraging* (averaging in proportion to the number of documents in each category). If we use both on a collection with an unbalanced distribution of documents into categories (like Reuters) then we could have two different views of the performance of the methods: the *microaverage* will reflect the behavior of the system on the document level, while the *macroaverage* will focus on the behavior on the individual categories, giving the same importance to the big and the small ones. The study of the variation of these two averages on the different systems and filtering measures can give us a deeper insight into the way they really act and into the dependence of their performances on the sizes of the categories. For this reason, we will use both and compare them.

4 Experiments

This section describes the corpus and shows the experiments over this corpus.

4.1 The Corpus

As in [8] and [12] we take the Reuters-21578 corpus. It contains short news related to economy published by Reuters during the year 1987 (it is publicly available at [6]). These stories are unbalanced distributed on 135 different prefixed categories. Each document is allowed to belong to one or more of these categories. We have chosen the split into train and test documents which was made by Apté (cf. [2]). After eliminating the documents with no body or no topics we have finally obtained 7063 train and 2742 test documents assigned to 90 different categories.

4.2 The Results

We have applied two different classification algorithms (ARNI and SVM) when local reduction feature according to the measures described is previously applied. A sweeping of filtering levels (fl) was made ranging from 20% to 98% (this level indicates the percentage of words of the vocabulary that are removed from the representation of the documents).

Since Reuters collection is unbalanced (there are categories with $1,000$ documents while others have only 2 documents) the behavior of *microaveraged* and *macroaveraged* F_1's can differ since *macroaveraged* F_1 gives the same *weight* to F_1 for each category while *microaveraged* F_1 assigns a *weight* to F_1 for each category proportional to the size of the category. Hence, let us study both measures.

Table 1 shows the *macroaveraged* F_1 for all filtering levels and feature reduction measures. These results show the improvement of F_1 when local feature reduction is performed for both classifiers, and all measures and filtering levels.

In addition, when *microaveraged* F_1 is studied (see table 2), it is easy to see that local feature reduction also improves the performance of both classifiers for all measures (except for D_{ir}) and for most filtering levels.

ARNI					
fl(%)	L_{ir}	D_{ir}	IL_{ir}	IG	tf
0% Local	78.08%	78.08%	78.08%	78.08%	78.08%
20%	79.04%	78.03%	79.69%	79.15%	78.99%
40%	82.16%	80.74%	78.65%	79.85%	78.93%
60%	85.29%	81.21%	80.46%	79.06%	78.13%
80%	86.37%	85.28%	80.02%	80.69%	78.67%
85%	83.79%	79.87%	77.69%	77.62%	80.06%
90%	84.24%	78.57%	82.01%	80.31%	79.62%
95%	81.89%	68.60%	82.65%	82.97%	83.05%
98%	77.94%	67.23%	84.92%	83.55%	79.50%
SVM					
fl(%)	L_{ir}	D_{ir}	IL_{ir}	IG	tf
0% Local	50.66%	50.66%	50.66%	50.66%	50.66%
20%	59.57%	57.74%	59.18%	58.58%	60.19%
40%	60.44%	56.55%	59.44%	58.56%	58.61%
60%	60.58%	55.15%	59.62%	59.97%	57.79%
80%	63.69%	57.28%	62.11%	61.75%	61.39%
85%	63.35%	56.83%	64.15%	63.40%	62.20%
90%	64.85%	58.16%	63.82%	64.68%	60.95%
95%	62.99%	58.51%	65.98%	65.34%	57.84%
98%	63.60%	58.39%	66.28%	64.30%	58.60%

Table 1. *Macroaverage* of F_1 for ARNI and SVM

However, the behavior of the systems varies with regard to the *microaveraged* and *macroaveraged* F_1's. In fact, if we select the best method according to *microaverage* we conclude that SVM is better than ARNI. However, if *macroaverage* is considered ARNI is clearly better. Something similar happens to the different measures. While the results of the *microaverage* suggest that IG should be used, with *macroaverage* L_{ir} seems to be a better choice (at least with ARNI).

These differences can be caused, as we exposed in section 3.4, by the different importance given by *macroaverage* and *microaverage* to the behavior of the systems on the individual categories. In Reuters, just two of the 89 categories (earn and acq) accumulate a half of the positive test examples. A system or measure which performs well on this two categories but not so well on the rest might have a bigger *microaveraged* F_1 than other that performs well on the other 87 categories but no so well on the big ones. Looking at the results obtained make us think that this might be happening here: SVM seems to perform better in presence of a large number of positive examples. With ARNI it seems to happen the other way round: it performs better than SVM on small categories but worse on the big ones. The same happens to the new introduced measures: they seem to work better than IG for small categories.

Then, we can raise an interesting conclusion. When dealing with unbalanced distributions of documents, a combination of systems and measures might improve the overall performance, and the size of the categories might be the key to take into account for the election of this combination.

ARNI					
fl(%)	L_{ir}	D_{ir}	IL_{ir}	IG	tf
0% Local	81.02%	81.02%	81.02%	81.02%	81.02%
20%	81.15%	78.82%	81.82%	82.32%	81.08%
40%	81.61%	78.37%	81.88%	82.47%	81.05%
60%	81.38%	78.92%	81.28%	81.91%	80.60%
80%	81.43%	78.84%	81.52%	81.63%	80.56%
85%	81.39%	78.97%	80.86%	82.18%	81.35%
90%	81.35%	78.73%	81.12%	81.77%	81.99%
95%	81.04%	77.32%	80.67%	82.01%	81.77%
98%	81.56%	78.30%	81.47%	82.21%	81.64%
SVM					
fl(%)	L_{ir}	D_{ir}	IL_{ir}	IG	tf
0% Local	84.87%	84.87%	84.87%	84.87%	84.87%
20%	85.03%	81.54%	84.92%	84.78%	84.86%
40%	85.16%	80.49%	84.99%	84.96%	84.76%
60%	85.26%	79.79%	84.67%	85.02%	84.76%
80%	85.21%	79.99%	85.07%	85.42%	85.00%
85%	85.05%	80.25%	84.89%	85.53%	85.03%
90%	84.81%	80.26%	84.53%	85.48%	85.28%
95%	84.47%	79.26%	84.35%	85.40%	84.74%
98%	83.43%	80.37%	83.69%	84.76%	83.38%

Table 2. *Microaverage* of F_1 for ARNI and SVM

5 Conclusions and Future Work

We have introduced some new measures for feature reduction based on concepts from ML. We have compared the performance of these measures and some classical ones (IG and tf) with SVM and ARNI on an unbalanced collection of documents which has been extensively used on the research of TC (Reuters).

The results of the different measures and of the systems vary when the *macroaverage* or the *microaverage* of the F_1's is considered. ARNI and the new measures obtain the best results with *macroaverage* and SVM and IG with *microaverage*. This suggests that there is a difference on the performance of the measures and the systems on big and small categories. Thus, we argue that the optimal results on TC when the collection of documents has an unbalanced distribution into categories might not be obtained with a single system and a single filtering measure. The size of the categories might have to be taken into account to choose and adequate system and filtering measure.

Then, it would be interesting to study further the influence of the size of the categories on the performance of the TC task with different algorithms and methods for feature reduction. When this is settled down it would be of great interest to develop a method capable of selecting *a priori* the best system and measure for each category depending on its size (and, possibly, on other factors).

References

1. D. W. Aha. *A Study of Instance-based Algorithms for Supervised Learning Tasks: Mathematical, Empirical, and Psychological Evaluations.* PhD thesis, University of California at Irvine, 1990.

2. C. Apte, F. Damerau, and S. Weiss. Automated learning of decision rules for text categorization. *Information Systems*, 12(3):233–251, 1994.
3. P. Clark and T. Niblett. The cn2 induction algorithm. *Machine Learning*, 3(4):261–283, 1989.
4. W. Cohen. Fast effective rule induction. In *International Conference on Machine Learning*, 1995.
5. W. Cohen. Learning to classify english text with ilp methods, 1995.
6. Reuters Collection. http://www.research.attp.com/lewis/reuters21578.html.
7. P. Domingos. Unifying instance-based and rule-based induction. *Machine Learning*, 24:141–168, 1996.
8. E. F-Combarro, I. Díaz, E. Monta nés, A. M. Pea, and J. Ranilla. Aplicacin de distintos mtodos de aprendizaje automtico a la clasificacin documental. In *Conferencia Iberoamericana en Sistemas, Ciberntica e Informtica CISCI 2002*, 2002.
9. J. Fürnkranz and G. Widmer. Incremental reduced error pruning. In *International Conference on Machine Learning*, pages 70–77, 1994.
10. T. Joachims. Text categorization with support vector machines: learning with many relevant features. In Claire Nédellec and Céline Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, number 1398, pages 137–142, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE.
11. S. Muggleton. Inverse entailment and progol. *New Generation Computing, Special issue on Inductive Logic Programming*, 13(3-4):245–286, 1995.
12. E. Monta nés, J. Fernández, I. Díaz, E. F. Combarro, and J. Ranilla. Text categorisation with support vector machines and feature reduction. In *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation CIMCA2003*, 2003.
13. M. F. Porter. An algorithm for suffix stripping. *Program (Automated Library and Information Systems)*, 14(3):130–137, 1980.
14. J. R. Quinlan. Constructing decision tree in c4.5. In *Programs of Machine Learning*, pages 17–26. Morgan Kaufman, 1993.
15. J. Ranilla, M. Garca-Pellitero, and A. Bahamonde. Construccin de rboles de decisin usando el nivel de impureza. In *Proc. of the VIII Conf. of the Spanish Asoc. for Artificial Intelligence*, volume I, pages 34–41, 1999.
16. J. Ranilla, R. Mones, and A. Bahamonde. El nivel de impureza de una regla de clasificacin aprendida a partir de ejemplos. In *VII Conferencia de la Asociacin Espaola para la Inteligencia Artificial, III Jornadas de Transferencia Tecnolgica de Inteligencia Artificial*, volume I, pages 64–71, Murcia, Spain, 1999.
17. J. Rochio. Relevance feedback in information retrieval. smart retrieval system. In *Experiments in automatic document processing*, pages 313–323. 1971.
18. G. Salton and M. J. McGill. *An introduction to modern information retrieval.* McGraw-Hill, 1983.
19. F. Sebastiani. Machine learning in automated text categorisation. *ACM Computing Survey*, 34(1), 2002.
20. M. R. Spiegel. *Estadstica.* McGraw-Hill, 1970.
21. V. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag, 1995.
22. T. Yang and J. P. Pedersen. A comparative study on feature selection in text categorisation. In *Proceedings of ICML'97, 14th International Conference on Machine Learning*, pages 412–420, 1997.

e e & A e e g
App f A d g e fe e e

s st l l s^2 t d z^2 d l d

a ora or o R N rona r c a S a a a o
D ar a n o cno og a a In or ac on a o n cac on
n r a a or ña
0 or ña S a n
{ ge t u i u e
tt // ti u e
Gr o S a Ra an D o ca ca a ac a ca
n r a San ago o o a
San ago o o a S a n
{ u i t y u e

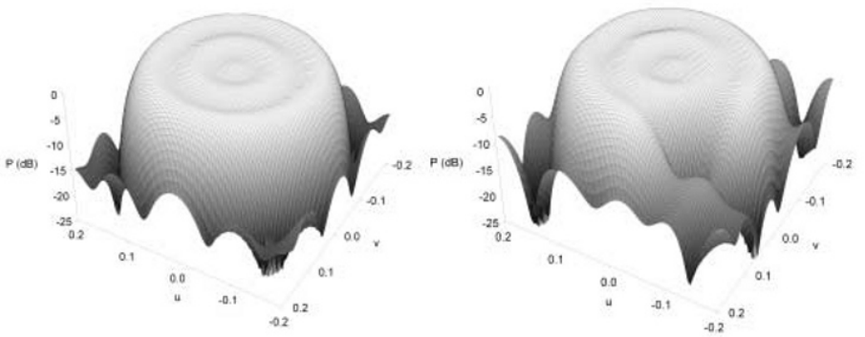
A st t r nc o n r r nc an a ro w n o
o wor ng w an nna In or r o a a r ff c
co on o n r c n rg n r c on o arr a o
nwan gna n ng on an nna con g ra on
roc o ca c a ng xc a on a w c a ow o o can
x r co x ana ca or o a on n r ca o
ar r q r roc a o o a o r a
corr c ac on ar c ro o o r c a N ra
N wor o a o n r r nc a ow ng o a corr c
a r a a o n a ar c

n uc i n

d s t s pl d t s t ll t s s t
st t l s t st t p t s pl d
s t ll t s
l s d s t t l ts pl d tw
d s l l tt t s l ts s d d ll t d w t
t st l t s w t s t s s p t s t ll t
t t st l s d t p w p tt s t st
t s p ss l t t l t z t t s
s
st t s s t p l s t t pp w
p pl w s w t t s l d w t pl s p t l
s t s t pl t l w s d l t l w t t
d l t s t d d t l t t s
t t s t s s d t t d sp p t t
s l t s sts s l t t t p t t d s ll p s t

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 750-757, 2003.

t p w p tt s ws w t d d d
t p s t w t s l s



F ow r a rn w an w o a q a n o on r n o a rn
can w n r no an n r r nc oc n r ar o n r r nc
can r c oca ng a q a n o on n ow r agra

t l ss t t p w p tt l s t ll
t t s t l s ll t t l ts t s t s d t
l l t t d p t s t w l
s l t s t s d t s t t d
t s t l ts t s s w s ll w
p s t d t p d d l d t s d s d t t s pp
s t t t ll t p ss l l p s t s st d
lt t s l t t t s d t s s w t p d d d t s
s sts t s *ul d l* t s st t
sp d s l t t s d t s w t s t s pp
t l s s t l p ss w t s t
l sp s t s t s ld d s l t l t t t
ff ts t t s t t t t d t t d
d w t t t d t st s t s d p l s
d pt d t s t l l tw s s s t
t p t s t t l ts t t t s t p tt s
p t tp t p sl s w
s s d s l ppl t s t ld t l t
t s p t l pl ppl t s s t st d d
st s w s d t p sp ts d t l w t
p tl t d d d d l d t p ts s s lt
t t d t st w ll d s w l s ss
pt d p tw p t s t ppl t pl
s p t d w t l l t d t sp t t

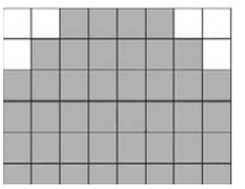
p l tp ts s d s p t p t s s t st p d t d s
t s tw t p s s t s l t pl s
t s S t s ld d s d t w s l t d
t l 6

M

t s s t s p ss l t p d s t pl s w l
t l t s tw t d t s d s d t st l s t tt t z
d t p t s t t l ts t t p ss s s
w ll l t t s l t t w ll s ds
t p t s t t l ts S w t s d t t d t
w ll p ss l t d t p w p tt l st st t sl d d
ts d s d ff ts

B he e

pl s d t s t l s p s d t d t
l ts t l ts d t d t
p t p t t w t s t l ts
s d w t t l ts s
s ws ts sp t l t w t s t t
l ts pl d t t s t s t s l ts s
p s d s d d l s S t t t l s l
l ts s 6 p ss t d d t *ul d l* s t
pt l t s d p l



F 2 annar arra a a con g ra on Gr q ar r r n ac
n

s p ss l t s t s t s ts l d l p d
w ll d t t t t p t s t s s
p s l l ts s w t t p t s w ll t

s ll p s t t p w p tt t p t d t s d s d
l p s t
st ll t s t d t d l d t s t
p t t p tt s s t d t t s ll p s t
d t p ts t w t t l s t t l ts
t s pp s s t d t d t
d l d t p ss s t
t d l d t s s t ll w t

l p s t s s d t l t t s ll p s t l s
^{o} d 6 ^{o} t t s t ppl d s
d

s s

v s s

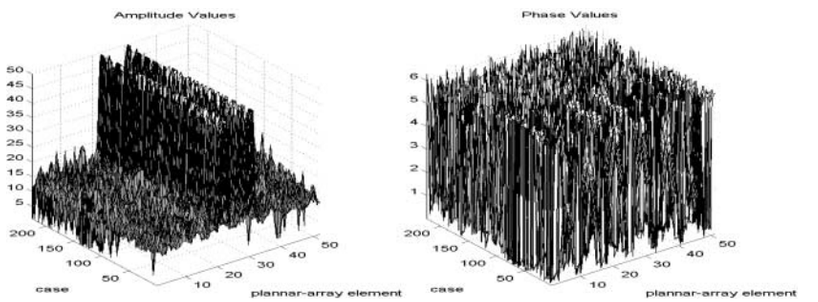
s s lt l s tw d t d
s pl s t d t s t s s s w
s l s w ll t s s d s p ts
tp t p s ts t t t l s l t s pl t d
d p s l s

t pt s d t d t p l w s t s s t *d l* p w p tt s w t p ts w t s d w z t s ll p s t s l d l t d S t s t d d s p t t p w p tt w t w w t t t l s t d t s s t s p tt s s p ts s t p ts t p w p tt t d t s tp ts w ll t l t t tw t s t

ppl d ff tw t st t l w st l t z s l w t p w t t s ll p s t s t t

p i n i n

st pp t s sts t t s d t s tp ts t sp d t t pl t d d p s l s t t t t s t s t t t s l s p s t d t t st d ff t s ts t tp t t w ll p ss l ts ff t t ls w d w t tw tw d ls d l s s tw s d d w d tw s s d s d ff t t t s w t d dd l s w t d s s d tw d d d ff t t s t s w s d l l t t s d d l t s d



F an a a I a n a row ar a n an o
ca or a or a o a n In ac gra r ar co n on or
ac o a rn

t t t t d t st s lts ll t s t ls w s
d w d tw w t dd l s d s sp t l d w t
t t s d t s t s t

t t p ss tw d ff t t p s s w s d
s S d p t t tp t l s
l l t d ll w l s d sp d t t
d l s t s d d tp ts t p tt s

$$\frac{\sqrt{\qquad}}{o \qquad o}$$

s tp ts t pl t d l s d t t t l s
t l ts t t t s lt t t d t w s
S 6 t s l t w s % t l d t s t t S
t d w s %

t w s t l t t t d t s p s d t sp t w t
t s l s p l t s pp t s t
t t d w s t s p ss
d t s t t d t s t w s spl t t tw p ts p ts t
t s t t d t s t s ll p s t t w
t tp ts t s ts w ll l t p s l s d t t s t
w ll t pl t d l s

s pp t d s s d l t t t
t ll ws t s lt s t t tw tw s

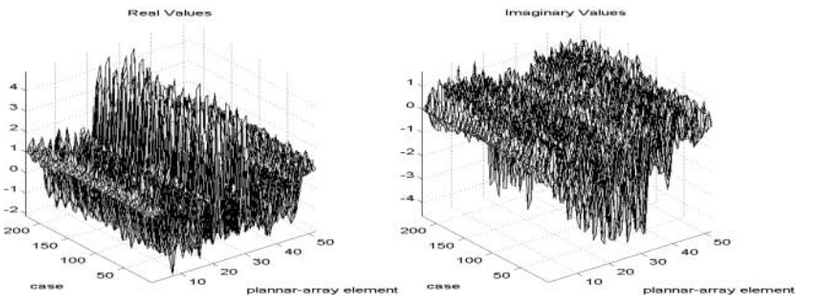
t t t t t s t t pl t d s t S w s
% t t s t d 6 % t l d t s t
t t t t t s t p s s t S w s 6 % d
% sp t ll s l s st ll d t t t sp t
t

t t t tw S l s l s t t d t sp d t l s l s t s s t t pl t d t s l s t tw d d t p s t s l s tw d t t s s lts p w p tt s s l t t d s d s t d t t t d ss ppl t p t t tt t st t s ppl d s t ll w t ff t l t t t s s t st s d t p s p d t t s s t p t w t w s w d t d st w t t d t p d t s s t p t d t d t s w l p tt t s tw pl s p s t s t s p t ll w t l z t l s t w s l d lt t p s t t t s l s st d t t t t pl t d s d p s s t t w s d d t t t l d p ts t t t t s t s s t s d s ws w t s s t t p s l d p ts ll w s t p tt tw t s l s l s t

R *A* s

A s

Us t s w p l d t d t t t S t d w s % t l p t d S % t p t t t l d t p tt s t S w s % t l p t d S 66% t s s t p w p tt s t d t tp ts t d t d t ppl d t s d t s ll l l

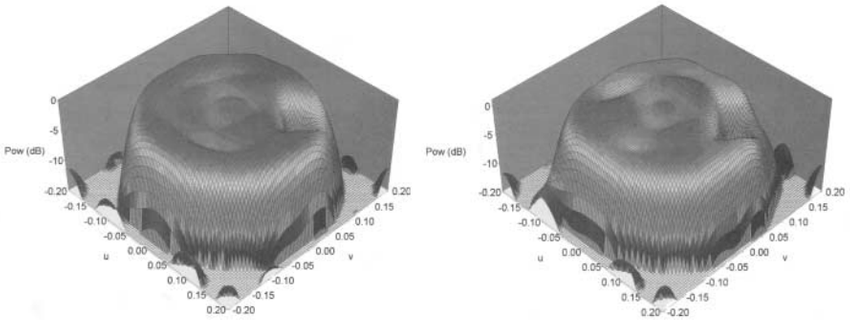


F R a an ag nar a

s t s p sl st t d t s lt t s l t t t p t d t s t s l d l s s tp ts st d s t d t s t t p t t t s ll t w s s t *d l*

p w p tt t t w w ld l t st l s t tp ts t

Us t s pp t t st s lts w t d s tw dd l s w t d s sp t ll t s s t t s t t S w s 6 % d 6% t l d p ts t t s t s t l d t s t t w s t d S % d % t t s l s t ppl s l w s ws t t t w d ff s tw t d s d tp ts d t s t d t t ls s t t t sp d d t s d



F 5 r ow r a rn a n w q a n o on a *u* 0 0 an *v* 0 0 an r 0 ow o co rag on I a a r o ±0 R g ow r a rn o a n ng NN o a q a n o on wa r 0 an r wa ±

l s ws t s p t s

rror n ra n ng an a a on a

			t o	
0 a	%		%	
a	%	0 %	0 %	%
R a I ag	%	%	%	%
ow r a rn R a I ag	%	%	%	0%

C nc usi ns

t t s w t s s w ff t d st t d t l t
t t s t s s t d ds t s d t s s w t
p w p tt s s t d t p d d p s t s w t st l s ll
t s t d s t tp ts st tl w ll ws t l
t t t t t t s d t t d

Ac n w n s

s w w s s pp t d t Sp s st S d l
d p ts d

f nc s

Ro r g J r no G or no S n o a G o a onar
n nna w g In n n ar a a I n nna an ro aga
on aga n 00
r o o G G org o o o ca on o N ra N wor n c
ro agn c r c o 00
So a S r J Donn D r c on n ng n a
rra w a N ra N wor a or r I ran ac on on n nna an
ro aga on aga n
Ra na a a ff J ra a an S g R a N ra
N wor S r c r w ca on o S ac cra S S on D c on
roc ng Jo n I S or ongr an 0 N I S In rna ona on
r nc anco r ana a 00
oog So a r o o x r n a a a on o a
N ra N wor D r c on n r I In rna ona S o on n nna an
ro aga on D g
oog r o o G G org o o a a In r
r nc anc a on n rc ar rra w Ra a a nc on N ra N wor
I In rna ona S o on n nna an ro aga on D g
0 0
rr In ro c on o Ra a a nc on N wo n r or ogn
Sc nc n r o n rg Sco an
a n S N ra N wor ac an o g ng o an

Feedback Linearization Using Neural Networks: Application to an Electromechanical Process

J.R. Alique[1], R. E. Haber[1,2], A. Alique[1]

[1]Instituto de Automática Industrial (CSIC).
km. 22,800 N-III, La Poveda. 28500 Madrid.
SPAIN.
{rhaber,jralique,a.alique}@iai.csic.es
[2] Escuela Técnica Superior
Universidad Autónoma de Madrid.
Ciudad Universitaria de Cantoblanco.
Ctra. de Colmenar Viejo, km 15. 28049 Madrid
SPAIN.

Abstract. This study explores the use of the feedback-linearization (FBL) paradigm using artificial neural networks (ANNs) to consider a force-control problem involving a complex electromechanical system, represented here by the machining process. The main goal is to control a single output variable, cutting force, by changing a single input variable, feed rate. Performance is assessed in terms of several performance measurements. The results demonstrate that the FBL strategy with ANNs provides good disturbance rejection for the cases analysed.

1 Introduction

During the last few decades, the principle of feedback linearization (FBL) has drawn much attention as a means of taking into account possible model inaccuracies in control systems [1]. The basic idea of FBL is to find a static-feedback control law such that the closed-loop system has linear input-output behaviour. Exact FBL involves the transformation of a nonlinear system into a linear one using a first-order differential equation that is usually quite difficult. However, the limitation of needing to have exact knowledge of the process's nonlinearities and the difficulties involved in performing nonlinear transformations have motivated the use of the FBL technique with artificial neural networks (ANNs) [2].

FBL with ANNs in control systems is still an active research area. Important contributions have been reported on incorporating a robustifying control term to ensure bounded control actions [3]. Moreover, a technique for designing a neurocontroller using feedback linearization is proposed in [4]. However, the process is not as easy as the authors claim when the model and its parameters are unknown or partially known and the transformation is difficult to carry out. Additionally, the utilization of feedforward networks within a direct model reference adaptive-control

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 758-765, 2003.

scheme using a novel internal model-control strategy is suggested in [5]. The most recent approach focuses on a generalized FBL [6]. Nevertheless, future research must incorporate the rejection of non-measurable disturbances and the automatic selection of parameters to achieve closed-loop stability.

One way to deal with complex electromechanical systems while preserving the benefits of linear-control techniques is to identify and cancel the nonlinearities using FBL [7]. In this paper a complex electromechanical process, the machining process, is used as the test bed to apply an FBL with ANNs [8]. The electrical portion of the system includes DC and AC rotational motors, amplifiers, sensors and other components. The mechanical portion includes the rigid structure and the body with its different shafts, gears and reducer. The main goal is to implement machining-process optimisation through controlling a single output variable, the cutting force, by changing a single input variable, the feed rate. A fixed neurocontroller designed on the basis of FBL technique is then used to control the linearized plant. The effectiveness of the FBL scheme is demonstrated through simulations and several performance criteria based on the given simulation results.

This paper is organized as follows: Section 2 introduces the machining-process models used in the simulations; Section 3 discusses ANN background; Section 4 addresses the design of a fixed neurocontroller using the FBL technique. Section 5 presents the simulation results. The final section draws a number of conclusions.

2 The Machining Process

The characteristics of the machining process as a complex electromechanical process severely limit the use of classic mathematical tools for modelling and control. The dynamics of the milling process (cutting-force response to changes in feed rate) can be approximately modelled using at least a second-order differential equation.

The following linear model is suggested in [9]:

$$G_{LU}(z)=\frac{0.019z+0.017}{z^2-1.75z+0.77}\,. \tag{1}$$

Another model obtained using second-order differential equations [10] is

$$G_{RS}(z)=\frac{0.052z+0.04}{z^2-1.42z+0.45}\,. \tag{2}$$

The structure of a first-order cutting-force process including cutting speeds and non-linear depth-of-cut effects is proposed [11] thus:

$$G_{LA}(z)=\frac{0.11}{z-0.85}a^{0.65}f^{0.63}\,. \tag{3}$$

$G_P=\{G_{LU}(z),G_{RS}(z),G_{LA}(z)\}$ represents the machining process from the classical viewpoint. Equations (1)-(3) are only valid over a narrow range; hence they cannot

trespass certain limits in representing the process' complexity and uncertainty. However, they do provide a rough characterisation of the dynamic behaviour of the machining process and are used in Section 5 for simulations.

3 Neural Network Background

Consider a nonlinear system with $\mathbf{u_k} \in R^m$ inputs and $\mathbf{y_k} \in R^p$ ($m \geq p$) outputs. Let us assume that the system can be exactly modelled by the following one-hidden-layer feedforward network.

$$y_{k+1} = \mathbf{W_o} \cdot \tanh(\mathbf{W_x} \cdot \mathbf{x_k} + \mathbf{W_u} \cdot u_k + \mathbf{b_x}) + \mathbf{b_o} \tag{4}$$

where $\mathbf{x_k} \in R^n$ is given by

$$\mathbf{x_k} = \left[y_k^{\mathrm{T}}, y_{k-1}^{\mathrm{T}}, \ldots, y_{k-ny}^{\mathrm{T}}, u_{k-1}^{\mathrm{T}}, \ldots, u_{k-nu}^{\mathrm{T}}\right]^{\mathrm{T}} \tag{5}$$

with $n = (ny+1) \cdot p + nu \cdot m$, $\mathbf{W_o} \in R^{p \times l}$, $\mathbf{W_x} \in R^{l \times n}$, $\mathbf{W_u} \in R^{l \times m}$, $\mathbf{b_x} \in R^{l \times 1}$ and $\mathbf{b_o} \in R^{p \times 1}$, where $\mathbf{b_x}$ is the bias vector in the hidden layer, $\mathbf{b_o}$ is the bias vector in the output layer, $\mathbf{W_x}, \mathbf{W_u}$ are input weight matrices and *tanh* represents the tangent hyperbolic function.

For the sake of simplicity, let consider the single-input single-output (SISO) nonlinear system ($y_k \in R^1$). The identification can be viewed as the determination of the mapping from the set $z^N = [\mathbf{u}\ \mathbf{y}]^{\mathrm{T}}$ to the set of possible weights (parameters) $\hat{\boldsymbol{\theta}}$ so that the network can produce a prediction $\hat{y}_{k+1}$ as close as possible to the actual output y_{k+1}

$$\hat{\boldsymbol{\theta}} = \left[vec(\hat{\mathbf{W}}_o)\ vec(\hat{\mathbf{W}}_x)\ vec(\hat{\mathbf{W}}_u)\ \hat{\mathbf{b}}_x\ \hat{\mathbf{b}}_o\right]^{\mathrm{T}} \tag{6}$$

Using a prediction-error identification method [12], the weights are calculated as

$$\hat{\boldsymbol{\theta}} = \arg\min\left(E(\boldsymbol{\theta}, z^{\mathrm{N}})\right). \tag{7}$$

A version of the Levenberg-Marquardt method was selected as the training algorithm [13].

The first step in using feedback linearization is to identify the system to be controlled. From the two general modelling structures [14], we can select the series-parallel model shown in (8) that expresses the approximation of the nonlinear process by the function $g(.)$ in terms of the past inputs and the past outputs of the system being modelled, the nonlinear autoregressive-moving average (NARMA) model:

$$\begin{aligned} \hat{y}(k+1|\boldsymbol{\theta}) &= g(\boldsymbol{\varphi}(k), \hat{\boldsymbol{\theta}}) \\ &= g(y_k, y_{k-1}, \ldots, y_{k-ny+1}, u_k, \ldots, u_{k-nu+1}, \hat{\boldsymbol{\theta}}) \end{aligned} \tag{8}$$

Using (4) to form the series-parallel model of (8), the model can be rewritten as (9) and (10).

$$\phi_{k(q)} = \tanh\left(\sum_{i=1}^{ny} a_{i(q)} y_{k-i+1} + \sum_{j=1}^{nu} b_{j(q)} u_{k-j+1} + b_{x(q,0)}\right) \tag{9}$$

$$\hat{y}_{k+1} = \sum_{q=1}^{p} W_{o(k,q)} \cdot \phi_{k(q)} + b_{o(k,0)} \tag{10}$$

In order to apply the input-output feedback-linearization explained in Section 3 below, affine models (i.e., models where inputs must appear linearly in the state-space description of the model) are necessary. The neural model given by (10) is not affine since the inputs do not appear linearly in the output. Considering an approximation of the NARMA model (8) called the NARMA-L_2 model in companion form,

$$\begin{aligned} y_{k+1} = {} & f_n\left(y_k, y_{k-1}, \ldots, y_{k-ny+1}, u_{k-1}, \ldots, u_{k-nu+1}\right) + \cdots \\ & g_n\left(y_k, y_{k-1}, \ldots, y_{k-ny+1}, u_{k-1}, \ldots, u_{k-nu+1}\right) \cdot u_k \end{aligned} \tag{11}$$

Equation (11) is an affine counterpart for (10) in compact form. Now, the main issue is to train two neural networks $f_n(\cdot), g_n(\cdot)$ with the same inputs to obtain an affine model of the dynamic system. The Levenberg-Marquardt training method can be used to determine the weights in $f_n(\cdot), g_n(\cdot)$. The derivative of the prediction with regards to the weights $\psi(k|\theta) = {\partial \hat{y}(k|\theta)}/{\partial \theta}$ is the key component in the implementation of the training method as well as the Hessian ${\partial \hat{y}^2(k|\theta)}/{\partial \theta^2}$.

Once again let us assume a single hidden MLP with tangent hyperbolic units in the hidden layer and a single-output linear neuron. The derivative of the overall model output (11) with respect to the weights consists of the derivatives of each network

$$\frac{\partial \hat{y}(k|\theta)}{\partial \theta} = \begin{bmatrix} \dfrac{\partial \hat{f}_n}{\partial \theta_{fn}} \\ \dfrac{\partial \hat{g}_n}{\partial \theta_{gn}} \cdot u(k-1) \end{bmatrix}. \tag{12}$$

4 Feedback Linearization with Artificial Neural Networks

First an ANN is trained to learn the dynamics of the process and is therefore given known input and output data sets. Training with real-time data uses data obtained from actual machining operations. According to (8) the dynamic equation can be described in reduced notation by

$$\hat{F}(t) = g(\mathbf{F}, \mathbf{v}) \ . \tag{13}$$

where g is an unknown function to be identified, F is the cutting force exerted during the removal of metal chips and v is the relative feed speed between tool and worktable. Cast in vector form, $\mathbf{v}$ and $\mathbf{F}$ are the input and output defined as $\mathbf{F} = [F(k-1) \ \cdots \ F(k-n)]$ and $\mathbf{v} = [v(k-1), \ \cdots \ , v(k-m)]$, k is the discrete time instant and n, $m \in Z$. A successful identification scheme should insure $\hat{F}(k)$ values as close as possible to those of $F(k)$ (actual output).

The training algorithm was developed using MATLAB and real-time data from actual machining operations. The topology was initially chosen as follows: one input v and one output $\hat{F}$, a linear activation function at the output and one hidden layer using the hyperbolic tangent for the activation function. The type of model was selected using *a priori* knowledge of the milling process and the types of models considered in previous work. An ANN with six neurons in the input, six neurons in the hidden layer and one neuron in the output layer was selected. The dynamic equation can be described in reduced notation by

$$\hat{F}(k+1) = g(v(k), \ F(k), \ F(k-1)) \ . \tag{14}$$

The use of feedback linearization using neural networks can be explained with an example of a single-input single-output dynamic system. Assume that it is known that the discrete model of the plant is a second-order model. In order to apply exact input-output feedback-linearization theory, affine models are necessary (i.e., models where v appears linearly in the description of the model):

$$\hat{F}_{k+1} = f_n\left(\hat{F}_k, \hat{F}_{k-1}, v_{k-1}\right) + g_n\left(\hat{F}_k, \hat{F}_{k-1}, v_{k-1}\right) \cdot v_k \ . \tag{15}$$

After obtaining $f_n(\cdot), g_n(\cdot)$ on the basis of the methodology described in Section 3 above, a feedback linearizing controller is yielded by computing the control action

$$v_k = \frac{F_r(k) - f_n\left(\hat{F}_k, \hat{F}_{k-1}, v_{k-1}\right)}{g_n\left(\hat{F}_k, \hat{F}_{k-1}, v_{k-1}\right)} \ . \tag{16}$$

Selecting F_r as appropriate linear combinations of past outputs plus a reference signal enables the closed-loop poles to be assigned. The input/output behaviour will be equal to the following second-order linear system

$$F_r(k) = 1.6 \cdot F_r(k-1) - 0.64 \cdot F_r(k-2) + 0.04 \cdot r(k) \ . \tag{17}$$

The result of model (15) for predicting cutting force in real time is shown in Figure 2. Several performance indices were used to evaluate the model. The prediction error (*NSSE*) was 0.37, the final prediction error (*FPE*) was 0.39 and the estimate of noise variance (*ENV*) was 0.77.

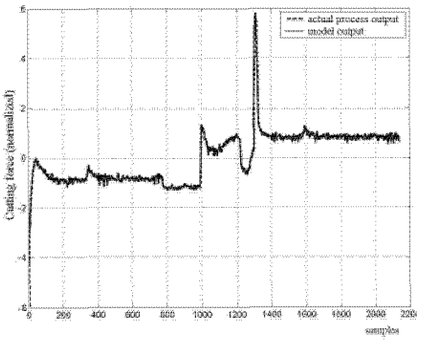


Fig. 2. Comparison between plant and model outputs

5 Simulations and Results

In the sequel, simulations were run based on linearized plant models (1)-(3) representing approximate process models and using the control scheme depicted in Figure 3.

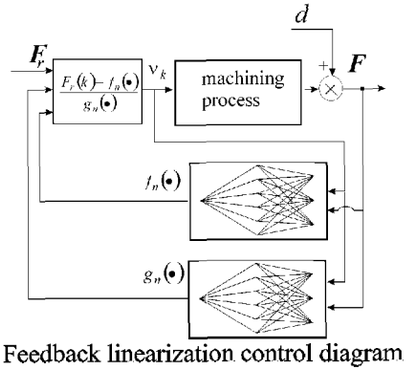


Fig. 3. Feedback linearization control diagram

In order to analyse the disturbance-rejection capabilities of the control system, additive noise plus the influence of unmodelled dynamics were considered in order to estimate the dynamics that can be expected in real-time applications. The following additive noise is assumed to corrupt the output:

$$d(t) = 0.1 \cdot (\sin 8t + \sin 12t + \sin 23.66t + \sin 35.49t) \,. \tag{18}$$

Now, the more realistic model of the process, including unmodelled multiplicative dynamics plus (21), is represented by

$$G_P(z) = G_*(z) \cdot \frac{0.095}{z - 0.904} + d(z) \,. \tag{19}$$

where $G_*(z)$ is an ideal process model represented by (1)-(3).

Figure 4 shows the simulation results with and without the influence of unmodelled dynamics and disturbances. In order to evaluate the simulation results, various performance indices were calculated such as integral absolute error (*IAE*), integral square error (*ISE*) and integral of time-weighted absolute error (*ITAE*). The overshoot, M_{pt}, was also computed.

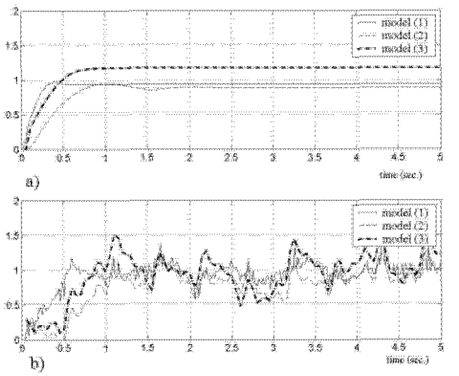


Fig. 4. Closed-loop response using FBL (a) without noise, (b) with noise.

Table 1. Control Strategy Comparison.

Model	*ISE*	*IAE*	*ITAE*	M_{pt}
(1)	4.85	19.95	36.64	-
(2)	6.88	22.54	65.08	-
(3)	6.57	28.16	100.23	17.72
(19)-(1)	12.96	35.68	63.40	38.61
(19)-(2)	15.22	33.15	81.92	30.01
(19)-(3)	14.97	35.45	115.81	56.61

Conclusions

This paper presents some preliminary results of a control strategy based on the feedback-linearization paradigm, which is a useful synergy of a dynamic ANN trained from real-life data and a traditional FBL scheme without the need for stable invertibility. The results shown here are still preliminary because only linear-process models have been considered for simulation. Nevertheless, the simulation tests show that FBL with ANNs performs quite well on the basis of *ISE*, *IAE* and *ITAE* performance measurements. Moreover, this control strategy is capable of stabilizing the plant in the presence of disturbances.

In this paper the ANN was trained off-line using actual data, but in future on-line training as well as the real-time application of the FBL will be implemented in order to enable adaptation and check actual plant behaviour.

References

1. Khalil, H., Nonlinear Control Systems, Prentice Hall, 2nd Edition, (1996)
2. Hagan, M.T., Demuth, H.B., de Jesus, O.: An Introduction to the Use of Neural Networks in Control Systems. International Journal of Robust and Nonlinear Control **12** (2002) 959-985
3. Yesildrek, A., Lewis, F.L.: Feedback Linearization Using Neural Networks. Automatica **31** (1), (1995) 1659-1664
4. He, S., Reif, K., Unbehauen, R.: A Neural Approach for Control of Nonlinear Systems with Feedback Linearization. IEEE Transactions of Neural Networks **9**(6) (1998) 1409-1421
5. Lightbody, G., Irwin, G.W.: Nonlinear Control Structures Based on Embedded Neural Systems Models. IEEE Transactions on Neural Networks **8**(3) (1997) 553-567
6. Goodwin, G.C., Rojas, O., Takata, H.: Nonlinear Control via Generalized Feedback Linearization Using Neural Networks. Asian Journal of Control **3**(2) (2001) 79-88
7. Buckner, G.D., Schuetze, K.T., Beno, J.H.: Intelligent Feedback Linearization for Active Vehicle Suspension Control: Journal of Dynamic Systems, Measurement and Control **123** (2001) 727-733
8. Haber, R.E., Haber, R.H., Ros, S., Alique, A., Alique, J.R.: Dynamic Model of the Machining Process on the Basis of Neural Networks: From Simulation to Real Time Application. Lecture Notes in Computer Science **2331** (2002) 574-583
9. Lauderbaugh, L.K., Ulsoy, A.G.: Model Reference Adaptive Force Control in Milling. ASME Journal Engineering of Industry **111** (1989) 13-21
10. Rober, S.J., Shin, Y.C.: Control of Cutting Force for Milling Processes Using an Extended Model Reference Adaptive Control Scheme. Journal of Manufacturing Science and Engineering **118** (1996) 339-347
11. Landers, R., Ulsoy, A.: Model-Based Machining Control. ASME Journal of Dynamics Systems, Measurement and Control **122**(3) (2000) 521-527
12. Ljung, L.: System Identification: Theory for the User: 2nd Edition, Prentice Hall, (1999)
13. Norgard, M., Ravn, O., Poulsen, N.K., Hansen, L.K.: Neural Networks for Modelling and Control of Dynamic Systems, Springer-Verlag, London (2000)
14. Narendra, K.S., Parthasarathy, K.: Identification and Control of Dynamical Systems Using Neural Networks. IEEE Transactions on Neural Networks **1** (1990) 4-27
15. Narendra, K.S., Mukhopadhyay, S.: Adaptive Control Using Neural Networks and Approximate Models. IEEE Transactions on Neural Networks **8** (1997) 475-485

Automatic Size Determination of Codifications for the Vocabularies of the RECONTRA Connectionist Translator*

Gustavo A. Casañ and M. Asunción Castaño

Dept. Ingeniería y Ciencia de los Computadores
Universidad Jaume I. Castellón, Spain.
{ncasan, castano}@icc.uji.es

Abstract. Previous work has recently shown that adequate and compact codifications of the lexicons involved in text-to-text MT tasks can be automatically created. The method extracted these representations from perceptrons with output contexts. They were later tested on a simple neural translator called RECONTRA. However, the size of the codifications was determined by hand using try-and-error mechanisms. This paper presents a method for automatically obtain such sizes by pruning the units of the hidden-layer of the perceptron encoder.

1 Introduction

Text-to-text limited-domain Machine Translation (MT) task has been recently approached using a simple neural translator called RECONTRA (REcurrent CONnectionist TRAnslator) [2] [4]. In this approach the vocabularies involved in the translations can be represented according to (simple and clear) local codifications. However, in order to carry out translations which involve large vocabularies through RECONTRA models with a non-excessive number of connections to be trained, distributed representations of both source and target vocabularies are required. A method to automatically create adequate and compact distributed codifications for the vocabularies in the RECONTRA translator was recently presented in [3]. The mechanism approached the problem through a Multilayer Perceptron (MP) in which output delays were included in order to take into account the context of the words to be encoded. However, the size of the distributed codifications was established by a human expert. This paper presents a method of automatically select the size of those codifications.

The rest of the paper is organized as follows: Section 2 describes the connectionist architectures employed to infer the lexicons representations and to translate the languages. The procedures used to train the architectures and the method used to extract the translations are also described. Section 3 presents the

* Partially supported by the Spanish Fundació Caixa Castelló, project P1·1B2002-15.

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 766-773, 2003.

task to be approached in the experimentation and Section 4 reports the translation performances obtained. Finally, Section 5 discusses the conclusions of the experimental process.

2 The RECONTRA translator and the codification generator

2.1 Network architectures

The basic neural topology of the RECONTRA translator is a simple Elman network [7]. Time delays are included in the input layer of this network, in order to reinforce the information about past and future events. Figure 1 illustrates this connectionist architecture.

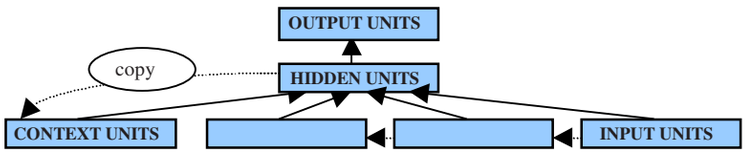


Fig. 1. Elman simple recurrent network with delayed inputs.

In order to automatically obtain the lexicons representations for the RECONTRA translator, several neural techniques can be employed. It could be convenient to obtain representations with similar codifications for words which have similar sintactic and/or semantic contexts. Taking this into account there are several possible methods using artificial neural networks, as Elman networks [6], RAAM (Recursive AutoAssociative Memory) machines [5] [10] or FGREP (Forming Global Representations with Extended backPropagation) [8].

The method adopted in this paper to encode the vocabularies uses a MP trained to produce the same oputput as its input (a word of the vocabulary to be encoded). The MP has as many input and output units as the number of words, since we use a local codification of the vocabulary. When the MP is trained enough, the activations of the hidden units have developed its own representations of the input/output codifications and can be considered the codifications of the words in the vocabulary. Consequently the size of the (unique) hidden layer of the MP determines the size of the distributed codifications obtained. In addition, a pruning method can be applied to this hidden layer to automatically determinate the size of these codifications.

In order to take into account the context in which the word appears, the corresponding previous and following words in a sentence are also showed at the output of the MP. Moreover, the importance of the input word over its context could be made equal, decreased or increased. Previous work on the matter [3] showed that better translation results were obtained with RECONTRA when the emphasis in the codification process was equated or put on the input word over its context; that is, when the input word of the MP encoder is repeated several times at the output window. Thus, a possible format of an output window of size 8 for an input word *x*

could be *x-2 x-1* ***x x x x*** *x+1 x+2*, where *x-2*, *x-1* are the two previous words of its context, and *x+1*, *x+2* are the two following words.

2.2 Training procedure

The MP encoder was trained to produce the same output word (and its context) as the word presented at the input layer. To this end, an on-line version of the Backward-Error Propagation algorithm [11] was employed. In addition, in order to obtain a possibly adequate size for the codifications of the vocabularies, a classical pruning method, called Skeletonization [9], was adopted. The method trained an initial MP for 10 presentations of the complete learning sample (epochs) and removed the hidden unit which led to the minor Mean Square Error (MSE) on the training set. Training continued with the resulting pruned MP. This prunning process was applied after each 10 epochs, until there were no more hidden units in the MP. Then, the topology of the pruned MP which had led to the least MSE was selected. The resulting topology of the MP coder was later trained for 3000 more epochs.

In the RECONTRA translator the words of every input sentence were presented sequentially at the input layer of the net. And the model should provide the successive words of the corresponding translated sentence, until the end of such output sentence (identified by a special word) was recognized. The model was trained through an on-line and truncated version of the BEP algorithm mentioned above [11]. With regard to the translated message provided by RECONTRA, the net continuously generated (at each time cycle) output activations. They were interpreted by assuming that the net supplied the output word for which the codification in the target lexicon was nearest (using the Euclidean distance) to the corresponding output activations.

For the training of both the MP and the RECONTRA, the choice of the learning rate and momentum was carried out inside the bidimensional space which they defined, by analyzing the residual MSE of a network trained for 10 epochs. The training process continued for the learning rate and momentum that led to the lowest MSE over the learning corpus. And the learning process stopped after a certain number of epochs (1000 epochs for the non-pruned encoder, 3000 for the pruned encoder and 500 for the translator). A sigmoid function (0,1) was assumed as the non-linear function. Context activations of RECONTRA (copied from the hidden layer) were initialized to 0.5 at the beginning of every input-output pair of sentences.

3 The experimental machine translation task: the Traveller task

The task chosen in this paper to test the method which automatically extracts codifications for the vocabularies in the RECONTRA translator was a subset of the Traveller MT task designed in the EuTrans project [1]. The task approaches typical situations of a Traveller at a hotel's reception in a country whose language he/she

does not speak. And the subtask includes (Spanish to English) sentences in which the Traveller notifies his/her departure, asks for the bill, asks and complains about the bill and asks for his/her luggage to be moved. The subtask had vocabularies of 178 Spanish words and 140 English words. In order to approach first a simpler task with smaller vocabularies, the grouping of some words and word sequences into categories was adopted. Specifically, two categories labeled by $DATE and $HOUR were included, which respectively represented generic dates and hours. This variant, that we called the categorized task, had 132 different Spanish words and 82 English words. The length of the non-categorized Spanish sentences ranged from 3 to 20 words, and the length of the English sentences, from 3 to 17. In the categorized task, the length of the categorized Spanish sentences ranged from 3 to 13 words, and the length of the English sentences, from 3 to 12. Some examples of both the categorized and non-categorized subtask are shown in Figure 2.

Spanish: ¿ Está incluido el recibo del teléfono en la factura ?
English: Is the phone bill included in the bill ?

Spanish: Me voy a ir el día $DATE a $HOUR de la mañana .
English: I am leaving on $DATE at $HOUR in the morning .

Fig. 2. Two pairs of sentences of the non-categorized and categorized Traveller tasks.

4. Experimental results

First, the categorized task was approached using MP encoders to obtain automatic codifications for the Spanish and English vocabularies with different hand-preestablished sizes. Afterwards, the pruning method was applied to the MPs to reduce the size of the codifications. Pruned encoders were also tried later to tackle with the non-categorized task. All the experiments were done using the Stuttgart Neural Network Simulator [12].

4.1 Training and test corpora

The corpora adopted in the translation tasks were sets of text-to-text pairs which consisted of a sentence in the source language and the corresponding translation in the target language. A learning sample of 5000 pairs of sentences and a test set of 1000 sentences were adopted. 3425 of the non-categorized task training pairs and 991 of the test pairs were different. In the categorized task 2687 pairs of the training sample and 771 pairs of the test set were different.

The corpus adopted in the training of the MP encoders were sets of text-to-text pairs, for both the categorized and the non-categorized task. Each pair consisted of an input word and the same input word together with its context (the preceding and following words in a sentence) as output. All pairs were extracted from sentences

which appeared in the training corpus employed for the corresponding translation task. All the repeated pairs extracted from the translation corpus appeared only once in the training set of the MP. Of course, the number of these pairs was different depending on the size of the output context used. If the context was zero, there were as many training pairs as words in the vocabulary, and as the context size increased, the number of pairs also increased. There were no test corpora for the codification process; it was indirectly evaluated later in the translation process.

4.2 Features of the networks

Previous experiments with RECONTRA on the Traveller task [Castaño,93] showed that 50 and 37 units were adequate to (manually) encode the words of the Spanish and English vocabularies of the categorized task, respectively; and 61 and 51 units for the Spanish and English vocabularies of the non-categorized task. In order to go further, in the experiments reported in this paper we tried to automatically encode each of the vocabularies of the categorized and non-categorized task with 25 units. Consequently, the non-pruned MP for coding the Spanish vocabulary which included categories had 132 inputs and 132 outputs (according to a local representation of the vocabulary), 25 hidden units and several (4, 6 or 8) output word delays; a MP with 82 inputs, 82 outputs, 25 hidden neurons and several (4, 6 or 8) output word delays was considered for the English vocabulary. If there was no right or left context of the input word, empty words were used instead. MPs with these features were also the initial networks for the pruning process.

For the Traveller task without categories, MPs with the same preestablished size of the hidden layer (25 units) and the same output context of the above encoders were used to begin the prunning process. However, in this case a different number of input/output units (178 and 140 for the Spanish and English vocabularies, respectively) were adopted.

The size of the above encoders determinated the size of the input and output layer of the RECONTRA translators adopted to approach the categorized and non-categorized Traveller tasks. In addition, taking previous experiments on the categorized task [4] as a reference, the corresponding translators had 140 hidden neurons and a window of 6 delayed inputs (with 2 words for the left context and 3 words for the right context).

As the task without categories was more complex and the sentences were longer, we increased the size of the input window of the RECONTRA translators adopted for the task to 8 words (3+1+4). The hidden layer size was increased to 180 units. Those values were suggested by the results obtained with previous experiments [4].

4.3 Results for the categorized Traveller task

The Traveller task with categories was approached using the MP encoders and the RECONTRA translators proposed for this task in Section 4.2. In a first experiment, both the Spanish and the English codifications were automatically obtained through

the non-pruned MPs (trained for 1000 epochs). Later, they were used to train the RECONTRA translators. Table 1 shows the test word error translation rates obtained, as well as the results of an experiment with (binary) manual codifications of the vocabularies.

Table 1. Test word error rates for the categorized Traveller task using manual and automatic codifications for the vocabularies

MP			Word Error Rate
| Output window|	Output window format	Size	
4	x-1 **x x** x+1	25	0.36
6	x-1 **x x x x** x+1	25	0.41
8	x-2 x-1 **x x x x** x+1 x+2	25	0.46
8	x-1 **x x x x x x** x+1	25	0.49
Distributed Manual codification		25	0.28

In a second experiment, the Skeletonization algorithm was applied to the MPs with output window of size 8 and 6 instances of the input word at the output. Two topologies with 12 and 11 hidden neurons were obtained for the Spanish and English encoders, respectively. Since the pruned MPs were smaller than the non-pruned networks, they were trained for 3000 epochs. Word codifications were extracted after 1000 and 3000 learning epochs and used in the RECONTRA translator. The test results obtained (which are showed in Table 2) reveal that very good rates were achieved.

Table 2. Test word error rates for the categorized Traveller task using pruned automatic codifications for the vocabularies

MP				Word Error Rate
| Output window|	Output window format	Size	Training epochs	
8	x-1 **x x x x x x** x+1	11/12	1000	5.68
8	x-1 **x x x x x x** x+1	11/12	3000	0.85

4.4 Results for the non-categorized Traveller task

Taking into account the encouraging results achieved in the previous sections, the Traveller task without categories was approached using only pruned MP encoders. The Skeletization algorithm was applied to different MP topologies, with the features described in Section 4.2. The resulting pruned MPs (with the number of hidden units showed in table 3) were trained for 3000 iterations. Then, word codifications were obtained. The RECONTRA translators proposed for this task in Section 4.2 were later trained and evaluated. Table 3 shows the test word error rates obtained, as well as the results of an experiment with (binary) manual codifications of the vocabularies. However, the translation results were not as good as they were expected.

Table 3. Test word error rates for the non-categorized Traveller task using manual and pruned automatic codifications and RECONTRA models with 180 hidden units

MP			Word Error Rate
|Output window|	Output window format	Size	
4	x-1 **x x** x+1	11/9	9.38
6	x-1 **x x x x** x+1	11/11	8.11
8	x-2 x-1 **x x x x** x+1 x+2	14/13	7.83
8	x-1 **x x x x x x** x+1	11/9	9.16
Distributed Manual codification		61/52	1.40

Later, the size of the hidden layer of the RECONTRA topology which had led to the best results in Table 3 (that in which the format of the output window of the MP was x-2 x-1 **x x x x** x+1 x+2) was increased up to 220 neurons. Nevertheless, the word error rates were only slightly decreased, as it can be observed in Table 4. The trained RECONTRA models were sometimes wrong in the translation of numbers (involved in days and hours) and months. In order to increase the distance between these words, two new codifications were considered, in which additional bits were included to the codifications previously obtained. First, the additional bits were manually included and the translation results were slightly increased, but the method stopped being completely automatic. In a second experiment, new MPs were trained (for 1000 epochs) to encode only the conflicting words, and the codifications extracted were later added to the codifications obtained before. The results, which are shown in Table 4, were as good as that obtained adding units manually. These results show the possibility of using this method for automatically creating representations for the RECONTRA translator.

Table 4. Test word error rates for the non-categorized task using pruned automatic codifications (extracted from MPs with output window x-2 x-1 **x x x x** x+1 x+2), with and without additional manual or automatically obtained units

	RECONTRA	MP		Word Error
	|Hidden layer|	Size	Training Epochs	Rate
Automatic codification	180	14/13	3000	7.83
	200	14/13	3000	6.07
Adding units manually	200	14/13 + 7/8	3000	4.53
Adding units automatically	200	14/13 + 5/5	3000 + 1000	4.38

5 Conclusions and future work

This paper proposes a method for automatically determining the size of the distributed codifications of the lexicons involved in a text-to-text MT task to be approached by the RECONTRA connectionist translator. The method extracts such

codifications of the hidden layer of a MP with output delays and the translation results achived are quite encouraging. The size of these codifications is determinated by an algorithm which prunes the hidden units of the MP. The translation results obtained with these automatic codifications can be sometimes improved by adding units to the codifications in order to increase the distance between similar words in the vocabulary.

The results showed in this paper reveal that those automatic codifications can still be improved. The addition of units to distinguish between similar words will continue being studied. Developing codifications for words and categories, and using them for categorized sentence translation, seems to be a more promising approach. On the other hand, more complex text-to-text MT task with larger vocabularies should be approached.

References

1. J. C. Amengual et al. *The Eutrans-I Spoken Language Translation System.* Machine Translation, vol. 15, pp. 75—102, 2000.
2. G. A. Casañ, M. A. Castaño. *Distributed Representation of Vocabularies in the RECONTRA Neural Translator*. Procs. of the 6th European Conference on Speech Communication and Technology, vol. 6, pp. 2423—2426. Budapest, 1999.
3. G. A. Casañ, M. A. Castaño. A*utomatic Word Codification for the RECONTRA Connectionist Translator.* Submitted to the 1st Iberian Pattern recognition and Image Analysis, 2003.
4. M. A. Castaño, F. Casacuberta. *Text-to-Text Machine Translation Using the RECONTRA Connectionist Model*. In "Lecture Notes in Computer Science: Applications of Bio-Inspired Artificial Neural Networks", vol. 1607, pp. 683—692, José Mira, Juan Vincente Sánchez-Andrés (Eds.), Springer-Verlag, 1999.
5. D. J. Chalmers. *Syntactic Transformations on Distributed Representations*. Connection Science, vol. 2, pp.53—62, 1990.
6. J. L. Elman. *Distributed Representations, Simple Recurrent Networks, and Grammatical structure*. Machine Learning, vol. 7, pp. 195—225, 1991.
7. J. L. Elman. *Finding Structure in Time*. Cognitive Science, vol. 2, no. 4, pp. 279—311, 1990.
8. R. P. Miikkulainen, M. G. Dyer. *Natural Language Processing with Modular Neural Networks and Distributed Lexicon*. Cognitive Science, vol. 15, pp. 393—399, 1991.
9. M. C. Mozer, P. Smolensky. *Skeletonization: A Technique for Trimming the Fat from a Network Via Relevance Assessment*. Advances in Neural Information Processing vol. 1, pp. 177--185, D. S. Touretzky, Ed. Morgan Kaufmann, 1990.
10. J. B.Pollack. *Recursive Distributed Representations*. Artificial Intelligence, vol. 46, pp. 77—105, 1990.
11. D. E. Rumelhart, G. Hinton, R. Williams. *Learning Sequential Structure in Simple Recurrent Networks*. In "Parallel distributed processing: Experiments in the microstructure of cognition", vol. 1. Eds. D.E. Rumelhart, J.L. McClelland and the PDP Research Group, MIT Press, 1986.
12. A. Zell et al. *SNNS: Stuttgart Neural Network Simulator. User manual, Version 4.1.* Technical Report no. 6195, Institute for Parallel and Distributed High Performance Systems, University of Stuttgart, 1995.

Integrating Ensemble of Intelligent Systems for Modeling Stock Indices

Ajith Abraham[1] and Andy AuYeung[2]

Department of Computer Science, Oklahoma State University, USA
ajith.abraham@ieee.org[1], wingha@cs.okstate.edu[2]

Abstract. The use of intelligent systems for stock market predictions has been widely established. In this paper, we investigate how the seemingly chaotic behavior of stock markets could be well-represented using ensemble of intelligent paradigms. To demonstrate the proposed technique, we considered Nasdaq-100 index of Nasdaq Stock Market[SM] and the S&P CNX NIFTY stock index. The intelligent paradigms considered were an artificial neural network trained using Levenberg-Marquardt algorithm, support vector machine, Takagi-Sugeno neuro-fuzzy model and a difference boosting neural network. The different paradigms were combined using two different ensemble approaches so as to optimize the performance by reducing the different error measures. The first approach is based on a direct error measure and the second method is based on an evolutionary algorithm to search the optimal linear combination of the different intelligent paradigms. Experimental results reveal that the ensemble techniques performed better than the individual methods and the direct ensemble approach seems to work well for the problem considered.

1 Introduction

Prediction of stocks is generally believed to be a very difficult task. The process behaves more like a random walk process and time varying. The obvious complexity of the problem paves way for the importance of intelligent prediction paradigms. During the last decade, stocks and futures traders have come to rely upon various types of intelligent systems to make trading decisions [1][3][4][5]. In this paper, we propose an approach to combine different intelligent paradigms using ensemble approaches to model the seemingly chaotic behaviour of two well-known stock indices namely Nasdaq-100 index of Nasdaq[SM] [9] and the S&P CNX NIFTY stock index [10]. Nasdaq-100 index reflects Nasdaq's largest companies across major industry groups, including computer hardware and software, telecommunications, retail/wholesale trade and biotechnology. The Nasdaq-100 index is a modified capitalization-weighted index, which is designed to limit domination of the index by a few large stocks while generally retaining the capitalization ranking of companies. Similarly, S&P CNX NIFTY is a well-diversified 50 stock index accounting for 25 sectors of the economy [10]. It is used for a variety of purposes such as benchmarking fund portfolios, index based derivatives and index funds. The CNX Indices are com-

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 774-781, 2003.

puted using market capitalisation weighted method, wherein the level of the Index reflects the total market value of all the stocks in the index relative to a particular base period.

Our research is to investigate the combination of the four different connectionist paradigms (using an ensemble approach) [6] for modeling the Nasdaq-100 and NIFTY stock market indices so as to optimize the performance indices (different error measures, correlation coefficient and so on). The four different techniques considered are an artificial neural network trained using the Levenberg-Marquardt algorithm, support vector machine, difference boosting neural network [11] and a Takagi-Sugeno fuzzy inference system learned using a neural network algorithm (neuro-fuzzy model) [4]. We analysed the Nasdaq-100 index value from 11 January 1995 to 11 January 2002 and the NIFTY index from 01 January 1998 to 03 December 2001. For both the indices, we divided the entire data into almost two equal parts. No special rules were used to select the training set other than ensuring a reasonable representation of the parameter space of the problem domain [3]. The trained connectionist paradigms were tested and the ensembles were integrated using two approaches. In Section 2, we briefly describe the different connectionist paradigms and the proposed ensemble approaches followed by experimentation setup and results in Section 3. Some conclusions are also provided towards the end.

2. Connectionist Paradigms

Connectionist models "learn" by adjusting the interconnections between layers. When the network is adequately trained, it is able to generalize relevant output for a set of input data.

2.1 Artificial Neural Networks

The artificial neural network (ANN) methodology enables us to design useful nonlinear systems accepting large numbers of inputs, with the design based solely on instances of input-output relationships. When the performance function has the form of a sum of squares, then the Hessian matrix can be approximated to $H = J^T J$; and the gradient can be computed as $g = J^T e$, where J is the Jacobian matrix, which contains first derivatives of the network errors with respect to the weights, and e is a vector of network errors. The Jacobian matrix can be computed through a standard backpropagation technique that is less complex than computing the Hessian matrix. The Levenberg-Marquardt (LM) algorithm uses this approximation to the Hessian matrix in the following Newton-like update:

$$x_{k+1} = x_k - [J^T J + \mu I]^{-1} J^T e \qquad (1)$$

When the scalar is zero, this is just Newton's method, using the approximate Hessian matrix. When is large, this becomes gradient descent with a small step size. As Newton's method is more accurate, is decreased after each successful step (re-

duction in performance function) and is increased only when a tentative step would increase the performance function. By doing this, the performance function will always be reduced in each iteration of the algorithm.

2.2 Support Vector Machines (SVM)

The SVM approach transforms data into a feature space *F* that usually has a huge dimension. It is interesting to note that SVM generalization depends on the geometrical characteristics of the training data, not on the dimensions of the input space. Training a support vector machine (SVM) leads to a quadratic optimization problem with bound constraints and one linear equality constraint. Vapnik shows how training a SVM for the pattern recognition problem leads to the following quadratic optimization problem [12]

$$\text{Minimize: } W(\alpha) = -\sum_{i=1}^{l} \alpha_i + \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j k(x_i, x_j) \tag{2}$$

$$\text{Subject to } \sum_{i=1}^{l} y_i \alpha_i \tag{3}$$

$$\forall i : 0 \le \alpha_i \le C$$

Where l is the number of training examples α is a vector of l variables and each component α_i corresponds to a training example (x_i, y_i). The solution of (2) is the vector α^* for which (2) is minimized and (3) is fulfilled.

2.3 Neuro-Fuzzy System

Neuro Fuzzy (NF) computing is a popular framework for solving complex problems [2]. If we have knowledge expressed in linguistic rules, we can build a Fuzzy Inference System (FIS), and if we have data, or can learn from a simulation (training) then we can use ANNs. For building a FIS, we have to specify the fuzzy sets, fuzzy operators and the knowledge base. Similarly for constructing an ANN for an application the user needs to specify the architecture and learning algorithm. An analysis reveals that the drawbacks pertaining to these approaches seem complementary and therefore it is natural to consider building an integrated system combining the concepts. While the learning capability is an advantage from the viewpoint of FIS, the formation of linguistic rule base will be advantage from the viewpoint of ANN. We used the Adaptive Neuro Fuzzy Inference System (ANFIS) implementing a Takagi-Sugeno type FIS [7].

2.4 Difference Boosting Neural Network (DBNN)

DBNN is based on the Bayes principle that assumes the clustering of attribute values while boosting the attribute differences. Boosting is an iterative process by which the network places emphasis on misclassified examples in the training set until it is correctly classified [11]. The method considers the error produced by each example in the training set in turn and updates the connection weights associated to the prob-

ability $P\ (U_m / C_k)$ of each attribute of that example (U_m is the attribute value and C_k a particular class in k number of different classes in the dataset). In this process, the probability density of identical attribute values flattens out and the differences get boosted up. Instead of the serial classifiers used in the AdaBoost algorithm, DBNN approach uses the same classifier throughout the training process. An error function is defined for each of the miss classified examples based on it distance from the computed probability of its nearest rival.

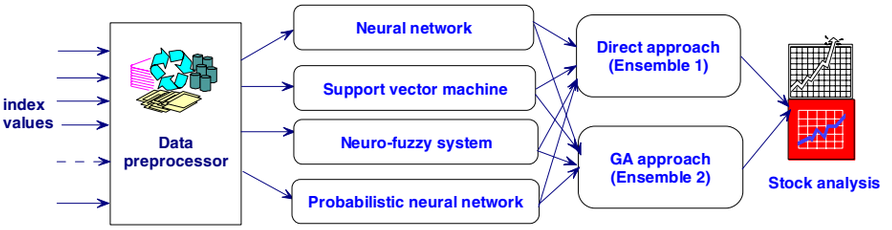


Figure 1. Ensemble approach to combine intelligent paradigms for stock modeling

2.5 Ensemble of Intelligent Paradigms

Optimal linear combination of neural networks has been investigated and has found to be very useful [6]. The optimal weights were decided based on the ordinary least squares regression coefficients in an attempt to minimize the mean squared error. The problem becomes more complicated when we have to optimize several other error measures. In addition to the Root Mean Squared Error (RMSE) and Correlation Coefficient (CC), we attempted to optimize the Maximum Absolute Percentage Error (MAP) and Mean Absolute Percentage Error (MAPE)

$MAP = \max\left(\frac{\left|P_{actual,\,i} - P_{predicted,\,i}\right|}{P_{predicted,\,i}} \times 100\right)$, Where $P_{actual,\,i}$ is the actual index value on day i and $P_{predicted,\,i}$ is the forecast value of the index on that day.

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{\left|P_{actual,\,i} - P_{predicted,\,i}\right|}{P_{actual,\,i}} \times 100$$, Where N = total number of days.

The first step is to carefully construct the different connectional models to achieve the best generalization performance. Test data is then passed through these individual models and the corresponding outputs are recorded. Suppose the daily index value predicted by DBNN, SVM, NF and ANN are a_n, b_n, c_n and d_n respectively and the corresponding desired value is x_n. Our task is to combine a_n, b_n, c_n and d_n so as to get the best output value that maximizes the CC and minimizes the RMSE, MAP and MAPE values. We propose the following two ensemble approaches.

Ensemble 1 (*E-1*): Determine the individual absolute error differences (example, $|x_n - a_n|$) and get the output value corresponding to the lowest absolute difference.

$$\min |a_n - x_n|, |b_n - x_n|, |c_n - x_n|, |d_n - x_n| \quad (4)$$

Ensemble 2 (*E-2*). Using a Genetic Algorithm (GA) search the optimal values for the linear parameters *m, n, o* and *p* such that

$$m + n + o + p = 1 \text{ and } a_n \times m + b_n \times n + c_n \times o + d_n \times p \approx x_n \quad (5)$$

so as to minimize RMSE, MAP and MAPE values and maximize the CC.

The fitness function could be modeled as

$$\text{Minimize}(Z) = (\text{RMSE} + \text{MAP}^{0.1} + \text{MAPE}^{0.2}) \times (1 - \text{CC}) \quad (6)$$

3. Experimentation Setup and Results

We considered 7 year's month's stock data for Nasdaq-100 Index and 4 year's for NIFTY index. Our target is to develop efficient forecast models that could predict the index value of the following trade day based on the opening, closing and maximum values of the same on a given day. For the Nasdaq-100index the data sets were represented by the '*opening value*', '*low value*' and '*high value*'. NIFTY index data sets were represented by '*opening value*', '*low value*', *'high value'* and *'closing value'*. We used the *s*ame training and test data sets to evaluate the different connectionist models. More details are reported in the following sections. The assessment of the prediction performance of the different connectionist paradigms and the ensemble method were done by quantifying the prediction obtained on an independent data set.

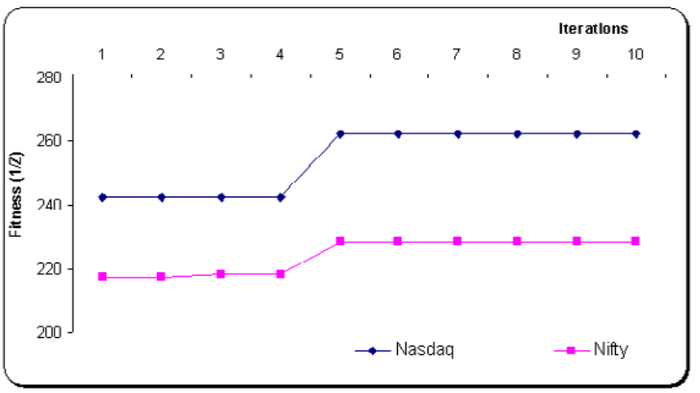


Figure 2. GA learning convergence using the ensemble (*E-2*) approach

- **Training of connectionist paradigms**

We used a feedforward neural network with 4 input nodes and a single hidden layer consisting of 26 neurons. We used tanh-sigmoidal activation function for the hidden

neurons. The training using LM algorithm was terminated after 50 epochs and it took about 4 seconds to train each dataset. For the neuro-fuzzy system, we used 3 triangular membership functions for each of the input variable and the 27 *if-then* fuzzy rules were learned for the Nasdaq-100 index and 81 *if-then* fuzzy rules for the NIFTY index. Training was terminated after 12 epochs and it took about 3 seconds to train each dataset. Both SVM (Gaussian kernel with = 3) [8] and DBNN took less than one second to learn the two data sets [3].

- **Parameter settings for the genetic algorithm**

Initial population was randomly created with the parameter settings as shown in Table 1. Each chromosome was represented using a 128 bits string having 32 bits for *m, n, o* and *p*. Figure 2 illustrates the GA convergence during the 10 iterations. Experiments were repeated 20 times for each data set and each trail run took about 4 seconds.

Population size	300
Iterations	15
Single point crossover and mutation	0.3 and 0.1
Selection strategy	Probabilistic

Table 1. Parameter settings of the genetic algorithm

- **Performance and results achieved**

Table 2 summarizes the training and test results achieved for the two stock indices using the four connectionist paradigms and the two ensemble approaches. Figures 3 and 4 depict the test results for the one-day ahead prediction of Nasdaq-100 index and NIFTY index respectively.

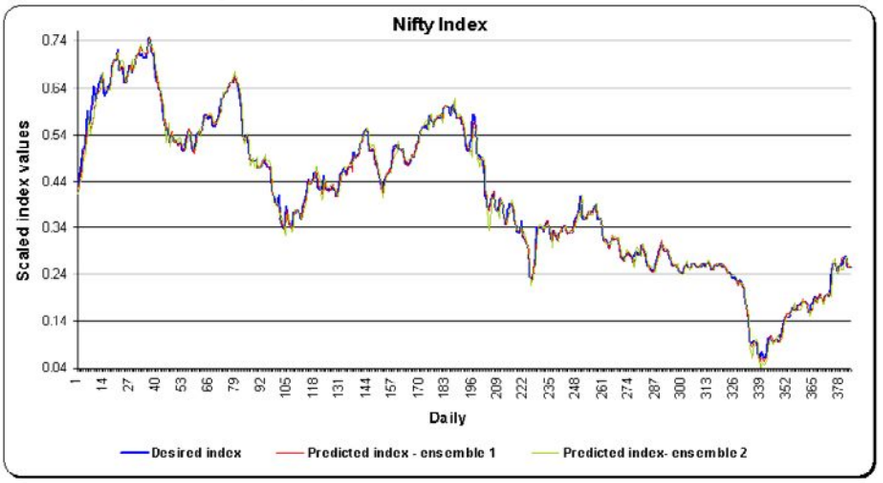


Figure 3. NIFTY index: performance of the different methods

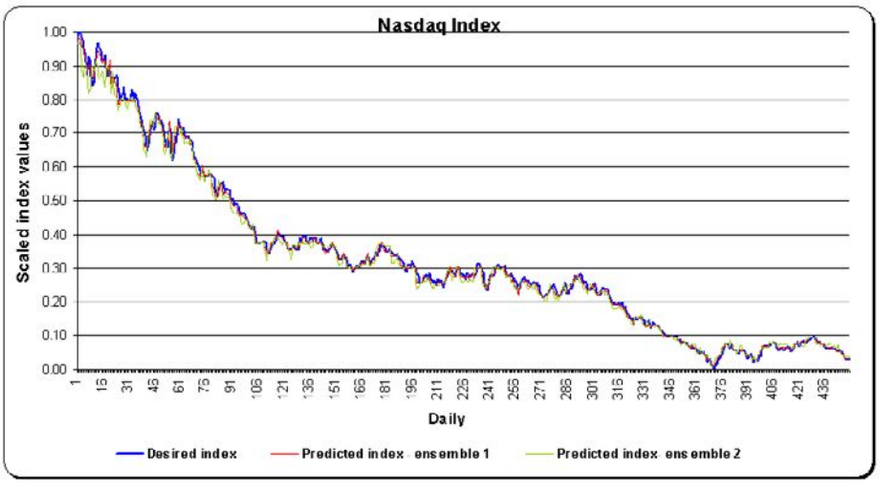


Figure 4. Nasdaq-100 index: performance of the different methods

Table 2: Empirical comparison of performance (training and test)

	SVM	NF	ANN	DBNN	E -1	E -2
Training results (RMSE)						
Nasdaq	0.0261	0.0221	0.0292	0.0292		
NIFTY	0.0173	0.0152	0.0143	0.0174		
Test results - Nasdaq						
RMSE	0.0180	0.0183	0.0284	0.0286	0.0121	0.0174
CC	0.9977	0.9976	0.9955	0.9940	0.9989	0.9979
MAP	481.50	520.84	481.71	116.98	94.20	436.3
MAPE	7.170	7.615	9.032	9.429	4.199	7.103
Test results - NIFTY						
RMSE	0.0149	0.0127	0.0122	0.0225	0.0081	0.0130
CC	0.9968	0.9967	0.9968	0.9890	0.9988	0.9969
MAP	72.53	40.37	73.94	37.99	23.62	64.070
MAPE	4.416	3.320	3.353	5.086	1.453	2.9049

4. Conclusions

In this paper, we have demonstrated how the chaotic behavior of stock indices could be well represented by ensembles of intelligent paradigms. Empirical results on the two data sets using the two ensemble approaches clearly depict the importance of the ensemble approach. It is interesting to note that the ensemble approach based on the direct error measurement (*E-1*) performed better than the GA approach. The output created by the *E-1* approach has the lowest RMSE, MAP, MAPE values and the highest correlation coefficient values for Nasdaq and Nifty indices. As depicted in

Table 2, E-2 approach could not optimize all the four objectives for the two problems considered.

Our research has clearly shown the importance of using ensemble approach for modeling stock indices. An ensemble helps to indirectly combine the synergistic and complementary features of the different learning paradigms without any complex hybridization. Since all the considered performance measures could be optimized such systems could be helpful in several real world applications. The developed E-1 ensemble was to predict accurately the index values for the following trade day based on the opening, closing and maximum values of the same on a given day. Our experimentation results indicate that the most prominent parameters that affect share prices are their immediate opening and closing values. The fluctuations in the share market are chaotic in the sense that they heavily depend on the values of their immediate forerunning fluctuations. Our study focus on short term, on floor trades, in which the risk is higher. However, the results of our study show that even in the seemingly random fluctuations, there is an underlying deterministic feature that is directly enciphered in the opening, closing and maximum values of the index of any day making predictability possible.

References

[1] Abraham A., Nath B. and Mahanti P.K., Hybrid Intelligent Systems for Stock Market Analysis, Computational Science, Springer-Verlag Germany, Vassil N Alexandrov et al (Editors), USA, pp. 337-345, May 2001.

[2] Abraham A., Neuro-Fuzzy Systems: State-of-the-Art Modeling Techniques, Connectionist Models of Neurons, Learning Processes, and Artificial Intelligence, Springer-Verlag Germany, Jose Mira and Alberto Prieto (Eds.), Granada, Spain, pp. 269-276, 2001.

[3] Abraham A., Philip N.S., Nath B. and Saratchandran P., Performance Analysis of Connectionist Paradigms for Modeling Chaotic Behavior of Stock Indices, Computational Intelligence and Applications, Dynamic Publishers Inc., USA, pp. 181-186, 2002.

[4] Abraham A., Philip N.S. and Saratchandran P., Modeling Chaotic Behavior of Stock Indices Using Intelligent Paradigms, International Journal of Neural, Parallel & Scientific Computations, USA, Volume 11, Issue (1&2), 2003.

[5] Francis E.H. Tay and L.J. Cao, Modified Support Vector Machines in Financial Time Series Forecasting, Neurocomputing 48(1-4): pp. 847-861, 2002.

[6] Hashem, S., Optimal Linear Combination of Neural Networks, Neural Network, Volume 10, No. 3. pp. 792-994, 1995.

[7] Jang J. S. R., Sun C. T. and Mizutani E., Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence, Prentice Hall Inc, USA, 1997.

[8] Joachims T., Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (Eds.), MIT-Press, 1999.

[9] Nasdaq Stock Market^{SM}: http://www.nasdaq.com.

[10] National Stock Exchange of India Limited: http://www.nse-india.com.

[11] Philip N.S. and Joseph K.B., Boosting the Differences: A Fast Bayesian classifier neural network, Intelligent Data Analysis, Vol. 4, pp. 463-473, IOS Press, 2000.

[12] Vapnik V. The Nature of Statistical Learning Theory. Springer-Verlag, New York, 1995.

Resolution of joint maintenance/production scheduling by sequential and integrated strategies

Fatima Benbouzid*, Christophe Varnier**, Nourredine ZERHOUNI**.

* Institut National d'Informatique (INI). BP 68M Oued Smar ALGER - ALGERIE.
EMAIL f_sitayeb@ini.dz
** Laboratoire d'Automatique de Besançon (LAB). 25, rue Alain Savary 25000 BESANCON France.
EMAIL {christophe.varnier, zerhouni}@ens2m.fr

Abstract. The article presents a comparative study on the strategies of scheduling unites maintenance/production in the workshops of the type flow shop of permutation. Two strategies are presented: the sequential strategy which resolution is done in two stages: initially we schedule the tasks of production then integrate the tasks of maintenance, taking the scheduling of the production as a strong constraint. The integrated strategy is the representation of the tasks of maintenance and production. The objective is to optimize an objective function which takes into account the criteria of maintenance and production at the same time. The Genetic Algorithms (AGs) proved their effectiveness in the scheduling of the production. A comparison among different heuristics implemented will conclude this work.

1 Introduction

Maintenance and production are two functions, which act on the same resources. However the scheduling of their respective activities is independent, and does not take account this constraint. These two elements having been established separately, their integration in the operation of the workshop poses a problem that is often solved by negotiation between the respective persons in charge of the two services in a sequential way.
We are thus against a multicriterion problem, at first to schedule the production under the constraints of respect of the deadlines, cost and quality of the products; on the second hand, to plan maintenance under the constraints of equipment reliability. This ensures the perenniality of the production equipment. The complexity of the problem due to uncertainties related in particular to the data and target makes us think that an approach by exact methods is not possible. There for we propose heuristic methods which make it possible to solve even partially this type of problem.

The article is devoted to the strategies of preventive maintenance and production joint scheduling in flow shop where each machine must be maintained periodically with intervals of known times. We used the genetic algorithms for their effectiveness proven in the scheduling of the production and in addition for the flexibility of representation of the individuals which lends itself perfectly to our problem. The aim being to optimize an objective function which takes into account the criteria of

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 782-789, 2003.

maintenance and production into same, the various results obtained for the various strategies are presented. Finally we will conclude with development prospects and extension for our work.

2 resolutions strategies

The interaction between maintenance and production, particularly their joint scheduling, is relatively little studied and rather recent in the literature [1, 2, 3, 4, 5, 6]. The majority of works concerning the Production/Maintenance relations use probabilistic approaches. The aim is to determine the best moment to plan a maintenance jobs according to a compromise between the maintenance cost and the risk of machines unavailability [7, 8, 9].
Binding together these two functions is natural. Effectively more and more maintenance small jobs are integrated in production scheduling. The objective is to plan the execution of the other maintenance jobs, changing the least possible the production plan and respecting the equipment maintenance periodicity.
One counts in the literature several strategies of joint scheduling. We will describe below the sequential and integrated strategies which aim is to solve conflicts between maintenance and production.

2.1 The sequential strategy

This strategy consists of two phases: the production scheduling followed by the integration of maintenance tasks that takes production as a constraint [10].

2.1.1 Production scheduling We will not explain here the operating mode of AGs; however, the reader can refer to Goldberg's work [11]. We present only the parameter setting which we chose.
Individual: The individual is a chain of N natural numbers, where each number identifies the number of the batch. The chain represents the sequence of execution of the batches in the workshop.
Evaluation of an individual: The *fitness* or function of adaptation of an individual measures its quality. We chose the function $f = 1/1+f1$ ($f1$ the sum of the delays) as a function of *fitness* the genetic algorithm tries to find a solution which minimizes this value.
Selection of an individual for the reproduction: The selection allows identifying the best individuals and duplicates them proportionally to their values of adaptation. It eliminates the weakest. One finds in the literature several operators of selection; we used the wheel of lottery, as well as the selection by row, and by tournament.
Selection of an individual for crossover: After the selection, one has a set of individual in a reproduction basin. We used two strategies of selection.
Crossover: It is the main operator of AG. Two individuals are combined to generate one, or two new individuals having the characteristics of the starting individuals. The chosen crossover is a k point crossover.

Mutation: It consists in permuting two positions chosen randomly in the individual. One generates for each individual a random number r in the interval [0,1]. If r < **mr** then the individual will be muted (with a mr rate). This rate controls the number of new individuals introduced into the population.

Stopping Criterion: The process of research is repeated until reaching the number of generations defined, or when one notices stagnation in the best solution after a certain number of iteration.

The genetic algorithm provides a scheduling of the production. This step is necessary for the following phase which consists in integrating the tasks of maintenance in these scheduling using two heuristics that we will present in the following paragraph.

2.1.2 Integration of maintenance tasks

Maintenance carried out is a systematic preventive maintenance where each task depends on the preceding one on the same machine (the date of beginning of each task of maintenance is dependent on the completion date of that which precedes it).

We propose initially two heuristics for the integration of the maintenance tasks in the case of only one machine. These heuristics will be the base of the resolution of the general problem of several machines (flow shop with maintenance).

The heuristics used in the case of only one machine are detailed in [12]. We consider the naïve and death search heuristics. We use two strategies for the flow shop: the up-down and the down up strategies.

The strategy Up_Down (UD)

One inserts all tasks of maintenance on the first machine using one of the heuristics developed for the problem with one machine. Then one goes to the second machine and so on to the last one. This strategy has the advantage that an inserted maintenance task will not be shifted.

The strategy Down_Up (DU)

In this case, one inserts all the tasks of maintenance on the last machine then on the one just before it and so on until reaching the first machine. The difference with the preceding strategy is that a task of maintenance which is inserted on a machine can be shifted thereafter, owing the fact that the insertion of a task of maintenance on a Mj machine is done after the insertion of all the tasks of maintenance on the machines Mi (I > J)

In the heuristic UD, the insertion of the tasks of maintenance is done independently, it is done without taking into consideration of already inserted tasks, as opposed to the heuristic UD.

2.2 Integrated strategy

This strategy consists on the common representation of maintenance and production tasks.

2.2.1 Coding of an individual Each individual is coded by a structure with two fields: the first field is a sequence S that represents the order of execution of the production jobs. The second is a matrix M that represents the sites of insertion of the maintenance jobs. The element M [i,j] represents the site of insertion of the j^{th} job of maintenance of i^{th} machine in the sequence S.

Example:

Production Sequence: S	9	8	5	3	1	2	0	7	4

Matrix M				
	0	1	4	6
	1	2	5	
	0	4	7	8

2.2.2 Reproduction Operators. A valid individual will be generated from two parents. This individual will inherit its information on the production and the maintenance of its parents. This leads us to define the following crossover operators:

1. The crossover on Production only.
2. The crossover on Maintenance only.

a- Horizontal crossover with even K points: The principle is as follows:

- generate K points randomly ∈ [0.. a number of machines].
- The sites of the maintenance jobs on the machines, which are in even parts of the first parent, are copied in the descendent 1, and the sites of the maintenances jobs that are in the remaining parts (odd) of the descendent 1 are copied from the odd parts of the second parent.

b- Horizontal crossover with K odd points: It is the same principle as the horizontal_crossover with K even points, except that in this case the odd parts are copied from the first parent and the even parts from the second.

The combination between the crossings of production and those of maintenance enables us to define others operators.

2.2.3 The mutation. The mutation can also be done on the production or maintenance. Three operators of mutation are proposed:

- Production mutation: it is done by generating two positions randomly, then changing the sites of the two production jobs which are with these two positions.
- Maintenance mutation: it consists in shifting randomly towards the left or the right-hand side one or more maintenance jobs.
- Vertical mutation: The principle of this operator is to permute the sites of the production jobs, which are in two different parts, by preserving the sites of the maintenance jobs that are inside of each part compared to the production jobs which are in the same part. Its effectiveness increases when the maintenance jobs are well placed compared to the production jobs.

3 Tests and Results

In this section, we will present a comparison between the results obtained by applying the various methods that we presented in the preceding section. We used Taillard Benchmarks to test our methods.

3.1 Resolution by the sequential strategy

The results obtained by applying the genetic algorithm on the benchmarks are summarized in table 1.

For the genetic algorithm, one notices that the number of points of crossover increases with the increase in the number of tasks of productions. For the problem (20*5 _ 2), the best solution is obtained at the iteration 198 out of 200. For the problem (50*5_1) it is obtained at iteration 298 out of 300, which proves the effectiveness of the operators of reproduction.

Table 1. Sequential genetic algorithm.

TA Benchmark	Main-tenance	Naïve heuristic			UD heuristic		DU heuristic	
		T *	T_{min}	T_{max}	Advance /Delay not allowed	Advance /Delay allowed	Advance /Delay not allowed	Advance/ Delay allowed
20*5-1	5-1	2801	3835	2124	2476	2264	2747	2632
20*5-2	5-1	4160	4826	2894	3266	3173	3365	3250
20*10-1	10-1	5383	10894	5109	4311	5034	5820	5684
20*10-2	10-1	6532	13923	5782	5672	5563	7019	6056
20*20	20-1	8557	15001	7086	3577	3967	5707	5608
50*5-1	5-1	17042	22439	14465	14658	13650	15779	16167
50*10-1	10-1	34539	67964	26003	28850	26665	30484	29225
50*20-1	20-1	54400	74859	36089	33065	33332	4523	43754
100*5-1	5-1	47193	78003	31614	36954	34003	41286	39482
100*10-1	10-1	119110	261583	82337	95268	92255	107537	98507
100*20-1	20-1	144130	233745	100405	89366	86083	129935	115476
100*10-1	10-1	537917	1114430	371950	517277	476828	521306	471737
200*20-1	20-1	662106	1059934	424350	498285	495428	587644	571600

The insertion of the tasks of maintenance on the sequences found by the genetic algorithm is done according to the naive heuristics (T *, T_{min} or T_{max}), the heuristics UD (by tolerating the advance and the delay of the tasks of maintenance or by prohibiting them) or the heuristics DU (by tolerating the advance and the delay of the tasks of maintenance or by prohibiting them).

We notice that the best solutions are obtained either by the Naïve heuristics (T_{max}) or by the heuristics HB. In table1, the performance of the heuristic Naïve (T_{max}) is better.

The execution of the two heuristic UD and DU with the algorithm which tolerates the advance and the delay of the tasks of maintenance provides better results that their executions with the algorithm which does not make it possible to advance or to delay the tasks of maintenance. This is due to the fact, that a task of maintenance has more possible sites in the first algorithm than in the second.

3.2 integrated strategy

The results of the resolution of the problem by the integrated genetic algorithm are given by the following table:

Table 2. integrated genetic algorithm.

Type benchmark	Integrated AGs									
	size	Nb	Select	sh	croisement				N°-géner	solution
					Prod	maint	Nbr	Select		
Ta20*5_1	50	700	Lot	+	P.O	-	10	Sel1	506	1874
Ta20*5_2	50	500	Rang	+	P.O	P-H	8	Sel1	186	2165
Ta20*10_1	50	500	Rang	+	P.O	P-H	12	Sel1	142	4003
Ta20*10_2	100	600	Rang	+	P.O	P-H	10	Sel1	326	3893
Ta20*20	100	100	Lot	+	P.O	P-H	12	Sel1	94	3602
Ta50*5	60	600	Rang	+	P.O	-	15	Sel1	493	12893
Ta50*10	60	600	Rang	+	P.O	-	15	Sel1	591	21981
Ta50*20	60	600	Rang	+	P.O	P-H	15	Sel1	580	33319
Ta100*5	60	800	Rang	+	P.O	P-H	15	Sel1	777	26130
Ta100*10	60	300	Rang	+	P.O	-	40	Sel1	246	80262
Ta100*20	40	1000	Rang	+	P.O	-	45	Sel1	999	86700
Ta200*10	40	200	Rang	+	P.O	-	25	Sel1	197	328984
Ta200*20	20	200	Rang	+	P.O	-	50	Sel1	197	460439

One notices that for the Ta20*20 problem, the solution was found with the iteration 94 sur100, for the Ta100*20 problem, it was found with iteration 999 out of 1000, and for the problems Ta200*10 Ta200*20 it was found with iteration 197 out of 200. That proof effectiveness ofthe operators of reproduction used.

4 Conclusion

The activities of production and maintenance act on the same resources. These two activities cannot, thus, be accomplished at the same time and on the same system, which can lead to forecast many conflicts in the use of the system by one or the other of the services related to maintenance or production. As a result, interest to develop a joint scheduling of these two activities emerged.

In this article, we studied the problem of the joint scheduling of the production and the preventive maintenance, in a workshop of production of the type *Flow Shop* of permutation. The goal is to optimize a multicriterion objective. This function is a compromise between the criteria of optimization of the production and those of maintenance. This problem being NP-complete, to solve it, we were interested in heuristics of general order such as the genetic algorithms, the NEH heuristic which proved their aptitude of resolution for a significant number of problems of optimization.

We have chosen to solve this problem using two strategies of scheduling, the integrated strategy and the sequential one. Concerning the sequential strategy, we proposed two heuristics for the scheduling of the production, the genetic algorithm and the adaptation of the NEH algorithm, and three other heuristics for the insertion of the tasks of maintenance. For the resolution by the integrated strategy, we developed two heuristics, the algorithm NEH and the genetic algorithms.

A comparative study among all the implemented methods showed that the integrated approach is the approach which gave the best results.

Improvements for this work can be seen under the following aspects:

- Improving the optimization methods by introducing Scaling in the genetic algorithms
- Changing the value of the parameters dynamically.
- Testing other hybridizations, such as hybridization between the production genetic algorithm and the hybrid genetic algorithm or the integrated genetic algorithm.
- Introducing other scheduling heuristics to confirm the conclusions of this work.

References

1. M.Ben-Daya & M.Makhdoum, Integrated production and quality model under various maintenance policies. Journal of the Operational Research Society, 49(8), pp.840-853, 1998.
2. M.Bennour, C.Bloch & N.Zerhouni, Modélisation intégrée des activités de maintenance et de production. 3^{e} Conférence Francophone de Modélisation et de SIMulation MOSIM'01 Troyes (France), 2 pp.805-810, 2001.
3. X.Qi, T.Chen & F.Tu, Scheduling the maintenance on single machine. Journal of the Operational Research Society, 50(10), pp.1071-1078 , 1999.
4. T.D.Rishel & D.P.Christy, Incorporating maintenance activities into production planning ; Intégration at the master schedule versus material requirement level. International Journal of Production Research, 34(2), pp.421-446, 1996.

5. P.Fraternali & S.Paraboschi, Ordering and selecting production rules for constraint maintenance: Complexity and heuristic solution. IEEE Transactions on Knowledge and Data Engineering, 9(1), pp. 173-178, 1997.
6. C.Y.Lee & Z.L.Chen, Scheduling jobs and maintenance activities on parallel machines. Naval Research Logistics, 47, p.145-165, 2000.
7. S.F.Smith, Knowledge-Base Production Management: Approaches, Results and Prospectives. Production Planning and Control, 4(3), 1992.
8. M.Brandolese, M.Fransi & A.Pozzeti, Production and maintenance integrated planning.
International Journal of Production Research, 34(7), pp. 2059-2075, 1996.
9. E.Sanmarti, A.Espuna & L.Puigjaner, Job production and preventive maintenance scheduling under equipment failure uncertainty. Computer Chemical Engineering, 21(10), pp.1157-1168, 1997.
10. F.Benbouzid, Y.Bessadi, S.Guebli, C.Varnier & N.Zerhouni, Résolution du problème de l'ordonnancement conjoint maintenance/production par la stratégie séquentielle. 4e Conférence Francophone de Modélisation et de SIMulation MOSIM'03 Toulouse (France), 2003.
11. Goldberg D.E., Genetic algorithms in search, Optimisation and Machine Learning. Addison-Wesley, Mass., 1989.
12. J.Kaabi, C.Varnier, N.Zerhouni, Heuristics for scheduling maintenance and production on a single machine. 2002 IEEE Conference on Systems, Man and Cybernetics. October 6-9, 2002 Hammamet, Tunisia.

MLP and RBFN for detecting white gaussian signals in white gaussian interference

P. Jarabo-Amores, R. Gil-Pita, M. Rosa-Zurera and F. López-Ferreras

Dpto. de Teoría de la Señal y Comunicaciones
Escuela Politécnica, Universidad de Alcalá
Ctra. Madrid-Barcelona, km. 33.600
28871, Alcalá de Henares - Madrid (SPAIN)
E-mail: {mpilar.jarabo,roberto.gil,manuel.rosa,francisco.lopez}@uah.es

Abstract. This paper deals with the application of Neural Networks to binary detection based on multiple observations. The problem of detecting a desired signal in Additive-White-Gaussian-Noise is considered, assuming that the desired signal samples are also gaussian, independent and identically distributed random variables. The test statistic is then the squared magnitude of the observation vector and the optimum boundary is a hyper-sphere in the input space. The dependence of the neural network detector on the Training-Signal-to-Noise-Ratio and the number of hidden units is studied. Results show that Radial Basis Function Networks are less dependent on the Training-Signal-to-Noise-Ratio and the number of hidden units than Multilayer Perceptrons, and approximate better the Neyman-Pearson detector.

1 Introduction

This paper deals with the application of Neural Networks (NN) to binary detection based on multiple observations. The problem of detecting a desired signal in Additive-White-Gaussian-Noise (AWGN) of zero mean and variance σ_n^2 is considered, assuming that the desired signal samples are also gaussian, independent and identically distributed random variables with zero mean and variance σ_s^2. This signal and interference model is one of the most used in the analysis and design of many communications and radar systems. The likelihood ratio detector (LRT) based on the Neyman-Pearson statistical hypothesis test is considered [1][2], whose performance relies on the knowledge of the interference and desired signal probability density functions (pdf). In actual situations, when the received signal characteristics are unknown and differ from those supposed in the model, the resulting performance of the LRT detector will be worse [3].

Neural Networks (NN) are proposed as a solution. Due to their ability to learn from their environment, and to improve performance in some sense through learning, NN can approximate the Bayessian optimum detector [4][5][6].

The objective of this work is the design of neural network based detectors, capable of approximating the Neyman-Pearson detector performance for white gaussian signals in white gaussian interference. Multi-layer Perceptrons (MLPs)

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 790-797, 2003.

and Radial-Basis-Function Networks (RBFNs) are considered. For analysing the performance of neural detectors, detection and false alarm probabilities can be estimated by means of conventional Monte-Carlo simulations. Due to the high computational cost of this method when simulating low probability events like false alarms, a modified Monte-Carlo simulation technique, called Importance Sampling (IS) has been used [7][8][9].

The influence of training parameters, like variance of signal and interference (σ_s^2 and σ_n^2), and network structure, is analyzed, in order to find the advantages and limitations of both detection schemes.

2 The optimum detector

Given a set of N observations, $\mathbf{z} = [z_1, z_2, ..., z_N]^T$, which define a point in a N-dimensional space, the detection system has to decide if they are originated either from noise only (hypothesis H_0) or from both noise and signal (hypothesis H_1).

To specify detector performance, the probability of detection (P_d) and the probability of false alarm (P_{fa}) are used. The probability of detection is the probability of deciding in favor of H_1 when it is the true hypothesis. The probability of false alarm is the probability of deciding in favor of H_1 when H_0 is true. When seeking a decision strategy that constraints P_{fa} to an acceptable value while maximizing P_d, the test is said to be a Neyman-Pearson Test [1], also called *"the most powerful test"* (MP), since it achieves the largest P_d among all the test that have the same P_{fa}.

If the observations under each hypothesis are gaussian, independent, and identically distributed random variables, the conditional probability density functions $f(\mathbf{z}/H_0)$ and $f(\mathbf{z}/H_1)$ are multivariate normal probability density functions. Assuming that under H_0, the variables $z_1, z_2, ..., z_N$ have zero mean and variance $\sigma_n^2 = 1$, while under H_1 they have zero mean with variance $\sigma_n^2 + \sigma_s^2 = 1 + \sigma_s^2$, the desired Signal-to-Noise Ratio (SNR) is defined in expression (1):

$$SNR = 10\log_{10}(snr) = 10\log_{10}(\frac{\sigma_s^2}{\sigma_n^2}) = 10\log_{10}(\sigma_s^2) \tag{1}$$

The likelihood functions $f(\mathbf{z}/H_0)$ and $f(\mathbf{z}/H_1)$ are expressed in (2) and (3), respectively.

$$f(\mathbf{z}/H_0) = \frac{1}{(2\pi)^{N/2}} exp(-\frac{1}{2}\mathbf{z}^T\mathbf{z}) \tag{2}$$

$$f(\mathbf{z}/H_1) = \frac{1}{(2\pi(snr+1))^{N/2}} exp(-\frac{1}{2(snr+1)}\mathbf{z}^T\mathbf{z}) \tag{3}$$

The decision rule in the Neyman-Pearson sense is expressed in (4):

$$\frac{f(\mathbf{z}|H_1)}{f(\mathbf{z}|H_0)} = \frac{1}{(snr+1)^{N/2}} exp\Big(\frac{snr}{2(snr+1)}\mathbf{z}^T\mathbf{z}\Big) \underset{H_0}{\overset{H_1}{\gtrless}} \eta_0 \tag{4}$$

where the threshold η_0 is determined from the specified value of P_{fa}. By taking logarithms on both sides and rearranging terms, (4) can be transformed into (5) showing that the test statistic is the squared magnitude of the observation vector $\mathbf{z}$.

$$\mathbf{z}^T\mathbf{z} \underset{H_0}{\overset{H_1}{\gtrless}} 2\frac{snr+1}{snr}\ln[\eta_0(snr+1)^{N/2}] \triangleq \eta \tag{5}$$

3 Neural network based detectors

In this approach, MLPs and RBFNs are designed for approximating the LRT detector based on the Neyman-Pearson statistical hypothesis test. The structures of the detectors are shown in figure 1. Both neural networks are finished with a hard threshold. If the output of the network is greater than the threshold, T, we decide in favour of hypothesis H_1. On the other hand, if the output of the network is lower than T, we decide in favour of hypothesis H_0.

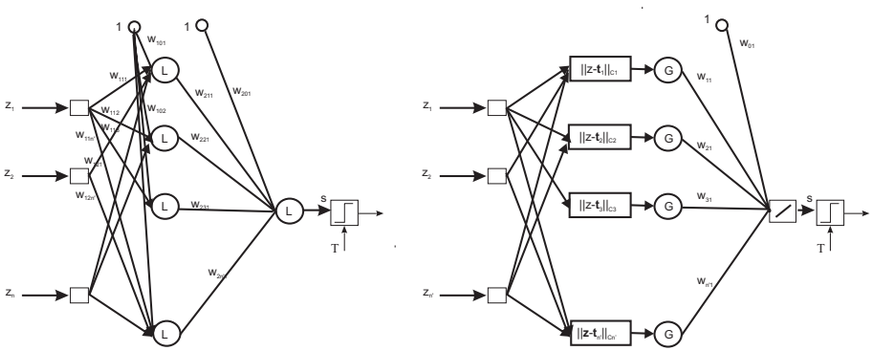


Fig. 1. MLP (left) and RBFN (right) structures

Cybenko's theorem [10] states that any continuous function $f : \mathbb{R}^n \to \mathbb{R}$ can be approximated with any degree of precision by sigmoidal functions (expression (6)). Because of that, MLPs with one hidden layer of neurons with sigmoidal functions have been considered.

$$L(x) = \frac{1}{1+exp(-x)} \tag{6}$$

In Radial-Basis Function Networks, the function associated to the hidden layer units (*radial-basis function*) is usually the multivariate normal function. The $i-th$ unit in the hidden layer is expressed in (7).

$$G_i(\mathbf{z}) = \frac{|\mathbf{C}_i|^{\frac{-1}{2}}}{(2\pi)^{\frac{N}{2}}} exp\{\frac{-(\mathbf{z}-\mathbf{t_i})^T\mathbf{C}_i^{-1}(\mathbf{z}-\mathbf{t_i})}{2}\} \tag{7}$$

where $\mathbf{C}_i$ is the covariance matrix, which controls the smoothness properties of the function. It can be set to a scalar multiple of the unit matrix, to a diagonal matrix with different diagonal elements or to a non-diagonal matrix. Equation (7) can be transformed into equation (8), making use of the weighted norm [11], $\|\mathbf{x}\|_C^2 = \frac{1}{2}\mathbf{x}^T\mathbf{C}^{-1}\mathbf{x}$:

$$G_i(\mathbf{z}) = \frac{|\mathbf{C}_i|^{\frac{-1}{2}}}{(2\pi)^{\frac{N}{2}}} \exp(-\|\mathbf{x} - \mathbf{t_i}\|^2_{\mathbf{C}_i}) \tag{8}$$

Taking into consideration the optimum decision rule expressed in (4), it is evident that a RBFN with only one hidden unit can implement the optimum LRT detector when the radial-basis function $G_1(\mathbf{z})$ is centered in the origin and its covariance matrix $\mathbf{C_1} = \mathbf{I}(\frac{snr+1}{snr})$, where $\mathbf{I}$ is the NxN identity matrix. In this sense, the output of the hidden unit is given by (9):

$$G_1(\mathbf{z}) = \frac{|\mathbf{C}_1|^{-1/2}}{(2\pi)^{N/2}} \exp\Big(-\frac{\mathbf{z}^T\mathbf{C_1^{-1}}\mathbf{z}}{2}\Big) = \frac{\Big(\frac{snr+1}{snr}\Big)^{-N/2}}{(2\pi)^{N/2}} \exp\Big(-\frac{(snr)\mathbf{z}^T\mathbf{z}}{2(snr+1)}\Big) \tag{9}$$

And the implemented decision rule is expressed in (10):

$$\Big(\frac{snr}{2\pi(snr+1)}\Big)^{N/2} \exp\Big(-\frac{snr}{2(snr+1)}\mathbf{z^T z}\Big) w_{11} + w_{01} \underset{H_0}{\overset{H_1}{\gtrless}} T \tag{10}$$

where $w_{11} = -(\frac{2\pi}{snr})^{N/2}(snr+1)^N$, $w_{01} = 0$, and $T = -1/\eta_0$. Such a RBFN only implements the optimum decision rule for a given value of snr. Looking for a RBFN capable of aproximating the optimum decision rule for a wide range of snr values, RBFNs with more than one hidden unit have been considered.

3.1 Description of experiments

MLPs and RBFNs for approximating the optimum detector in the Neyman-Pearson sense have been designed. In both cases, the neural network has N=16 input nodes and different number of hidden units, ranging from 8 to 56 in steps of 8, in order to study the performance dependence on network size. Another parameter that has been considered is the signal-to-noise ratio selected for training (TSNR). Different TSNRs, ranging from -3dB to 24dB have been used. For each TSNR value, separate training and validation sets composed of 50,000 randomly distributed patterns of only interference and signal-plus-interference have been generated for training the MLPs. For training the RBFNs, training sets composed of 5,000 randomly distributed patterns of only interference and signal-plus-interference have been generated. MLPs have been trained using the Back-Propagation algorithm, while a three-phases learning strategy [12][1] has been applied for training the RBFNs.

[1] In this strategy, the EM algorithm is used for fitting a mixture model to determine the centres of the RBFs, the BF widths are the maximum inter-centre squared distance and the weights from the hidden to the output layer are estimated using the LMS algorithm.

During training, the error between the network output (s in figure 1) and the desired output is calculated. After training, the relation between P_{fa} and detection threshold (T) has been estimated, presenting a new set of interference patterns (Monte-Carlo simulation). At this point, a new problem arises: how many interference patterns are needed for the relative error of the estimate of P_{fa} being lower than a given value? The relation between the number of interference patterns, M, the P_{fa} value to be estimated, and the relative error ε, is given in expression (11) [9]:

$$\varepsilon = \sqrt{\frac{1 - P_{fa}}{MP_{fa}}} \Rightarrow M = \frac{1 - P_{fa}}{\varepsilon^2 P_{fa}} \tag{11}$$

If $\varepsilon = 0.05(5\%)$ and $P_{fa} = 10^{-5}$, $M = 39.999.600$ interference patterns, and the computational cost incurred in Monte-Carlo simulation makes the estimate of P_{fa} quite complex. Importance Sampling (IS) is a modified Monte-Carlo technique based on altering the statistical properties of the input patterns in order to make those low-probability events (false alarms in our study) more frequent [7][8][9]. The introduced distortion is compensated by weighting the system output with a factor that is a function of the pdf of the interference patterns ($f(\mathbf{z})|H_0)$) and the pdf of the altered input patterns ($f^*(\mathbf{z})$) used for estimating P_{fa}. We have used an IS method based on the modification of the variance of the input patterns (Conventional Importance Sampling), which ensures an unbiased estimate of the P_{fa} in our case of study. For a given value of M, the minimum value of P_{fa} that can be estimated with an error lower than ε is different for each neural network (size and TSNR). For estimating the pairs (T,P_{fa}), and the associated error ε, we have used an efficient algorithm proposed in [9]. Due to the high number of analyzed networks, we have fixed the variance to 2.25 and $M = 10^6$ input patterns, that guarantee the relative error in the estimate of P_{fa} being lower than 5% in the results presented in subsection 3.2.

3.2 Results

Due to space limitation, only the most relevant results are presented. When analyzing the performance of the MLPs, the following conclusions can be extracted:

1. For a given network size, as the SNR increases, the difference between MLP performance and the optimum detector one decreases. The optimum detector performance is compared with the performance of MLPs with 32 hidden units in figure 2, for SNR=0dB and 7dB.
2. For a given SNR, as the number of hidden units increases toward 32, the MLP performance closes on the optimum detector one. For a number of hidden units higher than 32, no performance improvement is observed. Figure 3 shows P_d vs. P_{fa} curves for SNR=0dB and 7dB, and MLPs with 32, 40, 48, and 56 hidden units. Only the results obtained with the best TSNR value are presented, and only P_{fa} values lower than 10^{-5} are considered, because this is the case of interest in actual situations.

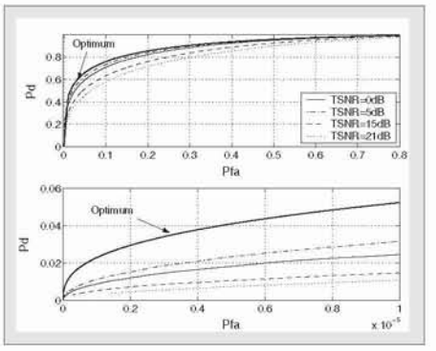


Figure 2.a.

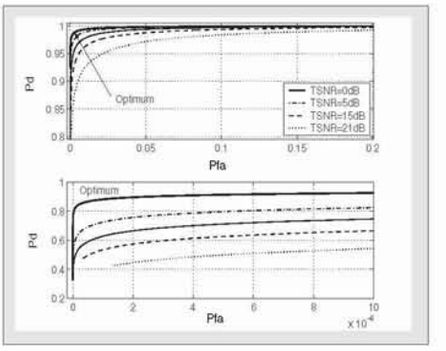


Figure 2.b.

Fig. 2. Optimum detector and MLP with 32 hidden units performances for SNR=0dB (figure 2.a) and SNR=7dB (figure 2.b)

Figure 4 shows the performance of RBFNs with 8 hidden units for SNR=0dB (figure 4.a) and SNR=7dB (figure 4.b). Results demonstrate that RBFNs with 8 hidden units trained with TSNRs lower than 11dB can approximate the optimum detector even for very low P_{fa} values. P_d vs. P_{fa} curves are presented for TSNR=0,5,17, and 21dB.

Due to the good results obtained with the RBFNs with 8 hidden units, a new set of RBFNs with 4 hidden units has been trained for the same set of TSNRs. Results demonstrate that 4 hidden units are enough for approximating the optimum detector. For TSNRs lower than 17dB, all curves approximate the optimum, and for higher values of TSNR, network performance is clearly worse. Figure 5 shows P_d vs. P_{fa} curves for SNR=0dB and SNR=7dB.

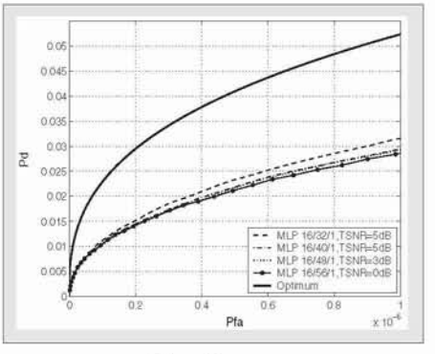


Figure 3.a.

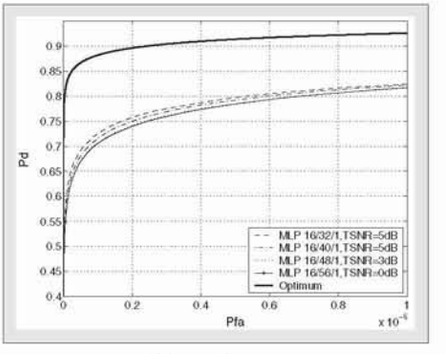


Figure 3.b.

Fig. 3. Optimum detector and MLP with 32, 40, 48, and 56 hidden units performances for SNR=0dB (figure 3.a) and SNR=7dB (figure 3.b)

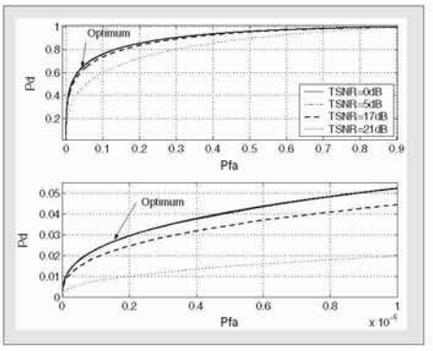


Figure 4.a.

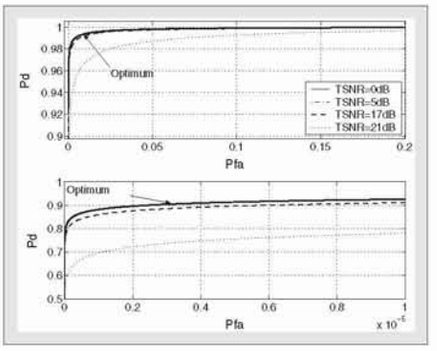


Figure 4.b.

Fig. 4. Optimum detector and RBFN with 8 hidden units performances for SNR=0dB (figure 4.a) and SNR=7dB (figure 4.b)

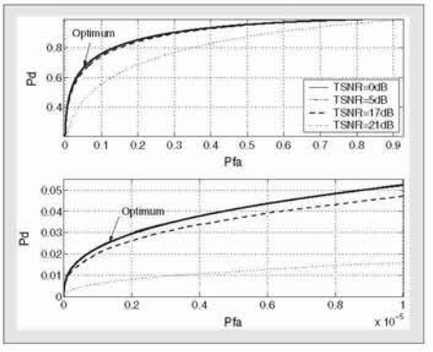


Figure 5.a.

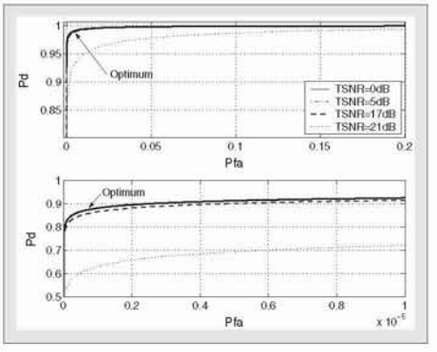


Figure 5.b.

Fig. 5. Optimum detector and RBFN with 4 hidden units performances for SNR=0dB (figure 5.a) and SNR=7dB (figure 5.b)

4 Conclusions

In this paper, the design of MLPs and RBFNs capable of approximating the Neyman-Pearson detector when considering white gaussian signals in white gaussian interference are considered. The dependence of these neural detectors on the TSNR and the number of hidden units is studied. Results show that RBFNs approximate the optimum detector better than MLPs.

When MLP are considered, the best performance is obtained with 32 units in the hidden layer, and no performance improvement is observed when the network

size is increased. On the other hand, a RBFN with only 4 radial basis functions can approximate the optimum detector for a wide range of TSNRs and very low P_{fa} values. This fact can be explained taking into consideration that the optimum boundary is a hyper-sphere in the input space.

References

1. M.D. Srinath, P.K. Rajasekaran, R. Viswanathan: Introduction to statistical signal processing with applications. Prentice-Hall, Inc., 1996.
2. V. Aloisio, A. di Vito, G. Galati: Optimum detection of moderately fluctuating radar targets. IEE Proceedings on Radar, Sonar and Navigation, Vol. 141, No. 3, pp. 164-170, June 1994.
3. A. di Vito, M. Naldi: Robustness of the likelihood ratio detector for moderately flutuating radar targets. IEE Proeedings on Radar, Sonar and Navigation, Vol. 146, No. 2, pp. 107-112, April 1999.
4. J. W. Watterson: An optimun multilayer perceptron neural receiver for signal detection. IEEE Trans. on Neural Networks, Vol. 1, No. 4, pp. 298-300, December 1990.
5. D.W. Ruck, S.K. Rogers, M. Kabrisky, M.E. Oxley, B.W. Suter: The multilayer perceptron as an aproximation to a Bayes optimal discriminant function. IEEE Transactions on Neural Networks, Vol. 1, No. 4, pp. 296-298, April 1990.
6. E.A. Wan: Neural network classification: A bayesian interpretation. IEEE Transactions on Neural Networks, Vol. 1, No. 4, pp. 303-305, December 1990.
7. G. C. Orsay: A note on estimating false alarm rates via importante sampling. IEEE Transactions on Communications, Vol. 41, No. 9, pp. 1275-1277, September 1993.
8. J. Grajal, A. Asensio: Multiparametric importante sampling for simulation of radar systems. IEEE Transactions on Aerospace and Electronic Systems, Vol. 35, No. 1, pp. 123-137, January 1999.
9. J. L. Sanz-Gonzalez, D. Andina: Performance anlisis of neural network detectors by importante sampling techniques. Neural Processing Letters, No. 9, pp. 257-269, 1999.
10. G. Cybenko: Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals and Systems, Vol. 2, pp. 303-314, 1989.
11. S. Haykin: Neural networks. A comprehensive foundation (second edition). Prentice-Hall Inc., 1999.
12. F. Schwenker, H.A. Kestler, G. Palm: Three learning phases for radial-basis-function networks. Neural Networks, Vol. 14, Issue 4-5, pp. 439-458, 2001.

Feature reduction using Support Vector Machines for binary gas detection

Maldonado-Bascón, S.[1], Al-Khalifa, S.[2], López-Ferreras, F.[1]

[1] Dpto. de Teoría de la Señal y Comunicaciones. Universidad de Alcalá, 28871 Alcalá de Henares (Madrid) Spain
[2] School of Engineering, University of Warwick, Coventry, CV4 7 AL. United Kingdom

Abstract. Gas sensor (electronic nose) has many different applications, such as fire detection, food quality control or medical application as well as the detection of atmospheric gases. We describe in this paper a signal processing technique using wavelet transform and Support Vector Machines (SVM) for CO and NO_2 gas detection and to obtain gas concentration. We propose a low complexity algorithm which can be implemented in a low cost palmtop gas monitor. SVM were used in a twofold way. First, SVM were used to classify the type of gas and then for the estimation of gas concentration.

1 Introduction

In this paper, a thermally-modulated sensor for gas sensing has been used. We use SVM applied to wavelet coefficients in order to obtain the type of gas and its concentration. The wavelet coefficients are calculated from the output sensor signal.

The gas sensor is based on the well-known advantages of using SnO_2. These advantages are its low manufacturing cost and high sensitivity. Its main disadvantages is a lack of long term stability and selectivity [1]. To overcome these disadvantages several techniques were used, such as operating at different temperatures. Temperature modulation has been reported as a recent way to enhance the selectivity of sensors. Gas sensors produced by employing micro-fabrication technology have been of much interest -mainly because of their potential use in array devices with low power [2]. Sensors, which are based on a micro-machined hotplate, have the extra advantage of a wide operating temperature range coupled with a rapid thermal time constant of around $1ms$. Micro-sensors also have the advantage of low cost because of batch fabrication. Our micro-sensor consists of an integrated heater-thermometer and sensing material. A large part of its production can be achieved by conventional foundries of CMOS chip. Periodic alteration between two temperatures has been considered for the improvement of sensitivity and selectivity and to reduce the effect of humidity. The micro-machined devices can be heated typically to around 500^oC and then cooled to room temperature in a few milliseconds. This will allow a fast control of the

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 798-805, 2003.

reaction kinetics of the oxide layer, which in turn produces a specific response pattern for a specific gas of interest.

The equivalent electric circuit of the sensor is showed in figure 1. The gener-

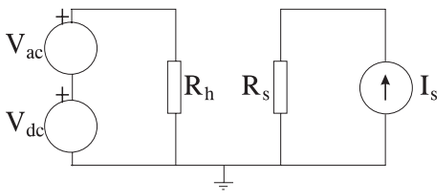


Fig. 1. Equivalent Electric Circuit of the Sensor

ated $V_{d.c.}$ controls the mean temperature component heating a platinum element and the modulation is produced with $V_{a.c.}$. Sampling the voltage response in the sensor resistance (R_s) excited by a constant current (I_s) a discrete time signal is obtained. The gas sensor is heated by a sinusoidal current and the resistance of the sensor varies with the concentration and type of gas.

2 Wavelet transform

Wavelet transform is an alternative technique to Fourier Analysis and the performance for non-stationary process is better than Fourier transform due to the time-scale localization properties, but even for stationary process the wavelet transform has very good localization properties. Due to the sinusoidal heating modulation, the resistance of the sensor varies in an alternate form. The response is sampled with 50mHz. One period of that signal is taken to be analyzed. The one period signal is extracted from the lowest level of the signal and the value of the first sample is subtracted in order to reduced transient at the beginning of the signal. Figure 2 shows the basic block diagram of wavelet transform, where the both linear time invariant systems $h[n]$ and $g[n]$ are 8 tap Daubechies filters [3], low pass and high pass respectively. Daubechies wavelets have a support of minimum size for any given number of vanishing moments. This basic scheme

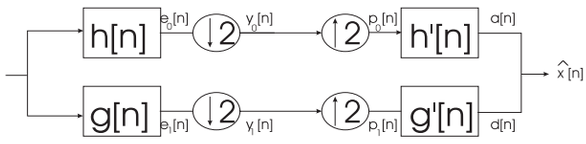


Fig. 2. Basic Wavelet Transform Scheme

are reiterated over low pass component four times as it is show in figure 3. The

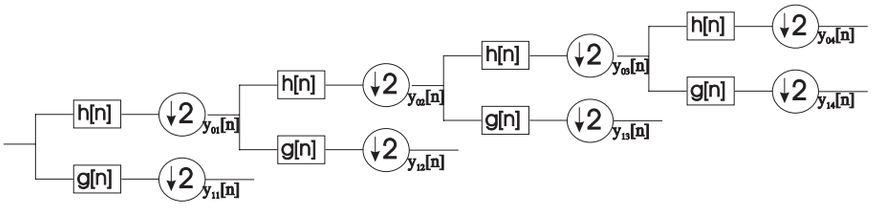


Fig. 3. Iterated Scheme

single period input signal is 25 samples long and the durations of the signals involved in figure 3 are those presented in table 1.

$x[n]$		25	
$y_{01}[n]$	17	$y_{11}[n]$	17
$y_{02}[n]$	13	$y_{12}[n]$	13
$y_{03}[n]$	11	$y_{13}[n]$	11
$y_{04}[n]$	10	$y_{14}[n]$	10

Table 1. Sequence Lengths

A single vector is formed with the signals y_{01}, y_{11},y_{02}, y_{12}, y_{03}, y_{13}, y_{04} and y_{14}, and the vector has 102 components.

In order to reduce computation time for a hand held system only those coefficients that are going to be used in the SVM for classification or regression are obtained. For this reason the equivalent system of every branch in figure 3 are calculated and the equivalent finite length impulse response filters are applied to the appropriate delay input signal.

This process is done using the noble identities, where the block showed in figure 4 is equivalent to the one shows in figure 5. The basic wavelet transform of figure 2 can be reduce to the scheme shows in figure 6, where $H_1(z) = H(z)$, $H_2(z) = G(z)$, $H_3(z) = H(z)H(z^2)$ and so on.

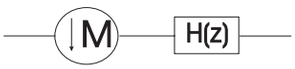


Fig. 4. Noble identities I

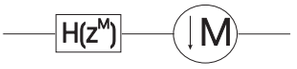


Fig. 5. Noble identities II

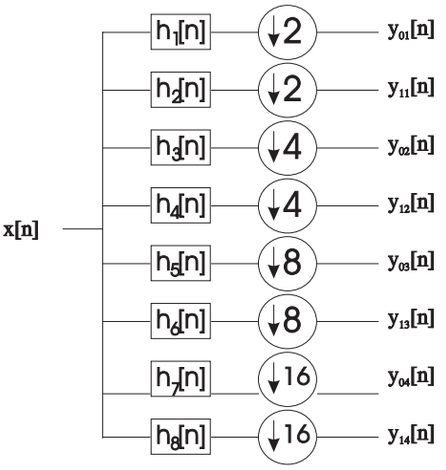


Fig. 6. Reduced wavelet transform scheme

3 SVM for gas detection

SVM can be used for classification, in our implementation three different gasses are considered, CO, NO_2 and a mix of CO and NO_2, and two SVM were used for classification. A SVM maps input data into a different space. In that new space an optimal hyperplane is obtained in order to separate two classes. Different mappings give different SVMs, namely $x \rightarrow \phi(x) \in H$, where x is the input vector (in the training process each input sample x_i has its associate class y_i, where two class are possible labelled by $y_i \in \{1, -1\}$, 1 for one class and -1 for the other class). The mappings $\phi(.)$ can be implemented in an implicit way using a kernel function $K(.,.)$ that defines an inner product in H. The decision of a SVM for a given input x depends on the sign of $f(x)$ where $f(x)$ is defined as:

$$f(x) = w\phi(x) + b = \sum_{i=1}^{N_s} a_i y_i K(x, x_i) + b$$

where N_s is the number of support vectors, w is the normal to the optimal plane, x_i are the support vectors and y_i is 1 or -1 depends on the input class. A detail description of SVM can be found in [4]. In our experiment, two linear SVM are using for classification. First, one class is CO and the other class is not CO (NO_2 or a mix of CO and NO_2). Then, when the gas is known not to be CO a second SVM is used to differentiate between NO_2 and the mix gas ($CO + NO_2$).

As we said, the wavelet transform block gives 102 coefficients, the most important features must be selected in order to reduce the computation time. In the linear SVM algorithm, the mapping function $\phi(x)$ is available, in fact there is not transformation.

$$\{\hat{x}_i, y_i\} = \{\phi(x)_i, y_i\} = \{x_i, y_i\}$$

The hyperplane used to classify is obtained by its normal w which is used to classify the input vector x. The angle θ_i of every component of the feature vector with the normal of the hyperplane can be obtained and an orthogonal feature component to w implies an irrelevant feature. Using this method, out of the 102 wavelet coefficients, the coefficients with the most information about the gas are selected. In figure 7 the magnitude of w are plotted from the result of the first classification linear SVM. There is a high correlation between adjacent coefficients. So we have selected the maxima from figure 7 in order to reduce the correlation between the selected coefficients to classify the gas. Next, the ten largest w components are taken as significant features from figure 7. The ten selected coefficients for the two classifications SVMs are showed in table 2.

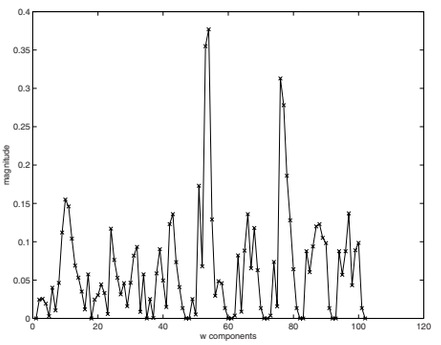


Fig. 7. Magnitude of w

	1	2	3	4	5	6	7	8	9	10
$1^{st}SVM$	$y_{01}[9]$	$y_{02}[8]$	$y_{12}[3]$	$y_{12}[6]$	$y_{03}[5]$	$y_{03}[7]$	$y_{13}[4]$	$y_{04}[5]$	$y_{14}[4]$	$y_{14}[7]$
$2^{nd}SVM$	$y_{01}[12]$	$y_{11}[3]$	$y_{11}[6]$	$y_{11}[12]$	$y_{11}[14]$	$y_{02}[9]$	$y_{12}[3]$	$y_{12}[5]$	$y_{12}[8]$	$y_{14}[5]$

Table 2. Selected coefficients for Classification SVM

The results of gas type classification using the reduce component vector are presented in table 3 and show that the gas is predicted with an accuracy of over 94%.

	CO	$notCO$
$1_{st}SVM$	100%	100%
	NO_2	$CO + NO_2$
$2_{nd}SVM$	100%	94%

Table 3. Classification accuracy results

4 SVM for gas concentration estimation

In $\epsilon-$SVM regression, our goal is to find a function $f(x)$ that has at most ϵ deviation from the actually obtained targets y_i for all the training data and at the same time is as flat as possible. The training data are given by $\{(x_1, y_1), (x_2, y_2), ..., (x_l, y_l)\} \subset \mathcal{X}\mathrm{x}\mathcal{R}$, here the values y_i are the values used for regress the function.

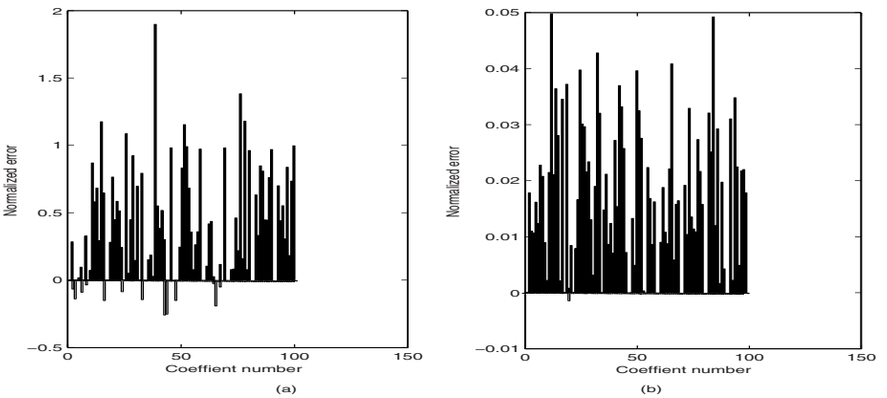


Fig. 8. Normalized relative error annulling one out 102 coefficients (a) CO (b) CO from the mix

SVM for classifications and regressions only depend on dot product between various patterns, the decision function can be written using a kernel as follow:

$$f(x) = \sum_{i=1}^{N_s} a_i y_i K(x, x_i) + b$$

where the hyperplane w can be no longer explicitly given. A detail tutorial in SV regression can be consult in [5]. Formally, we can write this problem as a

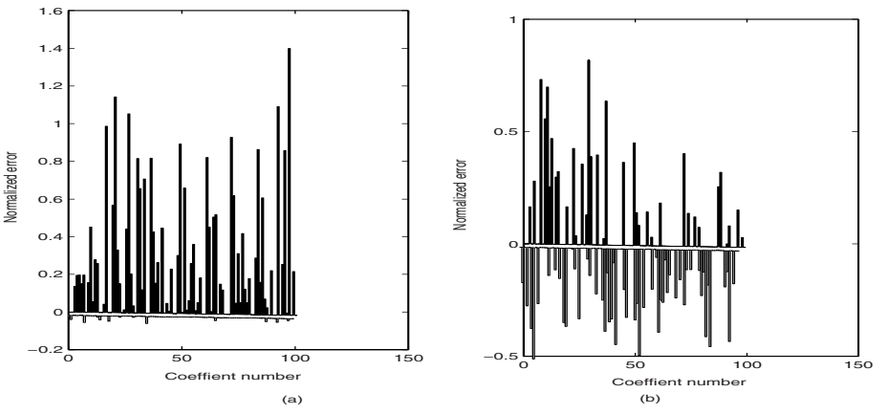


Fig. 9. Normalized relative error annulling one out 102 coefficients (a) NO_2 (b) NO_2 from the mix

convex optimization problem by requiring:

$$minimize \quad \frac{1}{2}||w||^2$$

$$subject \quad to \quad |y_i - f(x_i)| < \epsilon$$

A constant $C > 0$ determines the trade off between the flatness of f and the amount up to which deviations larger than ϵ are tolerated.

In our work the target values are the gas concentration. Four different SV regressions are used for each of the gasses we have: CO, NO_2, CO from the mix $CO + NO_2$ and NO_2 from the mix. A exponential kernel have been used for SV regressions. The input vectors are formed with the 102 coefficients from the wavelet transform. For each gas, the mean square relative error have been obtained for all the training data. We have repeated this process annulling one of the 102 coefficients and the normalized mean square relative error have been plotted (figures 8-9).

As can be see in figures 8-9, the error can be increased or decreased when one vector component is cancelling. For some vector components, the reduction in the error is big enough to propose to cancel it. In order to reduce the computation time, we propose to annul all coefficients that decrease the error or increase it below a threshold. The number of operations needed for regress one input is evaluated. The calculi of wavelet coefficients and the additions, multiplications and exponential from the decision functions must be taken into account.

The introduction of a new vector component is not very important due to the short impulse response of the equivalent filter of all filters in figure 6. The results from the four SV regressions are showed in table 4.

CO	
Number of vector components	15
Number of Support vector	56
Relative error	6.37%
NO_2	
Number of vector components	12
Number of Support vector	22
Relative error	4.57%
CO from the mix	
Number of vector components	22
Number of Support vector	48
Relative error	5.70%
NO_2 from the mix	
Number of vector components	47
Number of Support vector	29
Relative error	7.55%

Table 4. Results from SV regression for NO_2 from the mix

5 Conclusions

We have present in this paper the application of support vector machines for gas detection and to obtain gas concentration. Results show a high accurate for gas classification and relative error below 8% in all cases. The presented method is low computation time algorithm, so it can be implemented in a palmtop terminal with a low power consumption.

References

1. Ihokura, K.; Watson, J.: Stannic Oxide Gas sensors, principle and Applications. CRC Press, Boca Raton, FL (1994).
2. Gavicchi, R.E. et al.: Optimised temperature pulse sequence for the enhancement of chemicalle-specific response patterns from micro-hotplate gas sensor. Conf. Proc. Transducers '95 and Eurosensors IX. pp 823-826, Stockholm, 1995.
3. Mallat, S.: A wavelet tour of signal processing. Academic Press (1997).
4. Burges, C.J.C.: A tutorial on Support Vector Machines for pattern recognition. Data Mining and Knowledge Discovery. Vol. 2. N. 2 (1998) 121–167
5. Smola, A.J., Scholkopf, B.: A tutorial on Support Vector Regression. http://www.neurocolt.com

Artificial Neural Networks Applications for Total Ozone Time Series

Beatriz Monge-Sanz[1], Nicolás Medrano-Marqués[1]

[1] GDE - Área de Electrónica, Departamento de Ingeniería Electrónica y Comunicaciones,
Facultad de Ciencias, Universidad de Zaragoza
50009 Zaragoza, Spain
{bmonge, nmedrano}@unizar.es

Abstract. One of the main problems that arises when dealing with time series is the existence of missing values which have to be completed previously to every statistical treatment. Here we present several models based on neural networks (NNs) to fill the missing periods of data within a total ozone (TO) time series. These non linear models have been compared with linear techniques and better results are obtained by using the non linear ones. A neural network scheme suitable for TO monthly values prediction is also presented.

1 Introduction

During the last two decades, total ozone (TO) study has become an issue of high interest, not only for the climatological and meteorological communities, but also for biology, medicine and socioeconomics areas. The ozone layer situated in the stratosphere, approximately between 15-35 Km above the surface of the earth, acts as a shield that prevents solar shortwave radiation (UV), especially UV-B radiation (280-320 nm), from penetrating through the atmosphere to the surface. It is well documented that the reduction detected in ozone results in increasing levels of the harmful UV-B radiation at surface [1, 2]. That is one of the reasons that make the prediction of ozone levels a scientific matter of high interest. Ozone is usually expressed by a variable called total ozone (TO). TO for a specific location is the amount of ozone contained within the atmospheric column situated above the location. The most widely used units to measure ozone concentrations are Dobson units (DU) [1], one DU is defined to be equal to 0.01 mm thickness of ozone at standard temperature and pressure conditions (273 K and 1 atmosphere). At midlatitudes the average TO amount is about 300 DU, and it clearly exhibits two annual seasons: high TO during spring-summer and low TO during autumn-winter.

[1] Dobson units are named after G.M.B. Dobson (1889-1976), one of the first scientists to investigate atmospheric ozone. He designed the spectrometer used to measure ozone from ground, the Dobson Spectrometer.

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 806-813, 2003.

The science of ozone is quite complex, since many physical and chemical factors have an effect on its time variations. Some of the processes that affect TO are solar activity, chemical composition of the atmosphere, high/low pressure systems, troposphere-stratosphere energy exchange and atmospheric dynamics in general. All of them are extremely non-linear processes, which turns TO predictions into a more difficult task. The complexity of the problem makes neural networks ideal candidates to deal with TO data. Classical statistics assume the stationarity of the series, while neural networks structures are able to catch some temporal variation inherent to the series, for example a change of the relationship between the explained variable and the factors that cause its variations. On the other hand, conventional regression analyses make the analyser adopt a linear relationship between dependent and independent variables, whereas neural nets have no restrictions regarding the form of the relationship between input and output variables.

Section 2 of this paper offers a brief presentation of TO time series. In Section 3, two models based on neural networks (NNs) are presented for the substitution of missing values within the data series. In Section 4, the prediction of future values by using NNs is discussed.

2 Total Ozone Time Series

Extracting as much information as possible from observed time series is essential for progress in modelling and forecasting the system of interest. For this purpose it is desirable that the series do not contain missing values, but actually experimental series are seldom complete due to multiple causes. That is why previous to almost every analysis of the series a treatment for the missing data is required.

Depending on the nature of the series, sometimes it suffices to replace the missing data by the mean value of the whole series. Better options when the gaps are isolated are those based on persistence or linear interpolation of adjacent data. However, when dealing with atmospheric magnitudes, linear relationships are not very reliable mathematical fictions. Models based on neural networks are expected to give out more accurate results since they do not suppose linear dependence between variables and they are even able to track changes in the form of the relationships throughout the time. On the other hand, regarding the forecasting of TO amounts, neural networks are expected to play an important role when classical statistics are not able to provide a good enough response, or they do it in a complex or unreliable manner.

In this work we have chosen TO series corresponding to stations located in European mid-latitudes, which because of their geographical position are strongly influenced by atmospheric dynamics. All TO data have been retrieved from the World Ozone and Ultraviolet Radiation Data Centre (WOUDC). The time series used for this work are made up of monthly total ozone values for the following locations: Arosa in Switzerland, Vigna di Valle in Italy, and Lisbon in Portugal.

3 Missing Values Substitution

The artificial neural networks used here are based on feedforward configurations of the multilayer perceptron (MLP). Backpropagation is one of the most well-known training algorithms for MLPs, and it is indeed the preferred algorithm in most meteorological applications [3]. However, for this work the Levenberg-Marquardt [4, 5] method has been chosen because it overcomes two of the main problems that arise from backpropagation training: the Levenberg-Marquardt can converge fastly, and the risk for the final weights to be trapped in a local minimum is much lower, though it requires more computational memory [6].

Missing values within a time series of data could be grouped into isolated missing data and long gaps of missing values. For the estimation of isolated values, sometimes persistence based models or linear interpolation techniques are able to complete the series; we will show later that a NNs non linear model improves the accuracy of the results. The estimation of longer gaps requires more complex and specific methods and NNs are a promising alternative.

3.1 Isolated Missing Values

For the estimation of isolated gaps within the data series, the n previous and the n following values to the missing one are presented as inputs to the network. Best results are obtained for $n=2$, compared with those obtained for $n=1$, $n=3$ y $n=4$. The physical reason is fairly probably that the number of data considered with $n=2$ is 5 ($2+2$ predictors and the missing one), thus almost one semi-period of the annual ozone signal component is being taken into account. TO series, at the latitudes here considered, present a seasonal periodicity of 12 months, high ozone season for spring-summer and low-ozone season for autumn-winter. So taking fewer values we were not assuming variations within an interval affecting the searched value, and with more predictors we were including seasonal variations which must be avoided for the neural network to detect anomalies.

The structure of the network used for this interpolation model is [$2n$ $2n$-1 1]. For the single output neuron the lineal transfer function is used, while for the other layers the log-sigmoid function has been chosen. Data series were standardized (mean value equal to zero and unity standard deviation) before being processed by the net.

One of the goals of this work is to compare the accuracy of the results obtained with the non linear models with those given by linear interpolation techniques. In this case, linear interpolation has been calculated by using a perceptron with linear activation function that has been trained with the LMS method or Widrow-Hoff rule. This model generates the same results as a linear regression [7]. Simple statistics have been applied to estimate the confidence of the neural network model and to provide easy comparison with the lineal interpolation model. The chosen statistics tests have been: the standard error of estimate (RMSE), which is the positive square root of the mean squared error, and the variance explained (VE) which is simply the square of the correlation coefficient between the original series and the output of the model; the VE

informs about the amount of variation within the original series that can be explained by the model. Simple validation has been performed for the models, using the set of data June1967-November1973 for training the net, and the remaining part for validation. Figure 1 shows the results for the period December1973-July1975 for the three stations. For the non linear model the considered resulting series are the average of ten runs. The corresponding RMSE and VE per cent values are summarized in Table 1. The value of the VE is higher than 99.90 % for both linear and non linear models, however the RMSE clearly shows the improvement achieved by the NNs non linear model. For the non linear interpolation, the RMSE is slightly lower than that of the linear model for the series of Arosa, significantly lower for Vigna di Valle and more than six times lower for the series of Lisbon. Therefore it is shown that the non linear model improves the results obtained with a linear regression model.

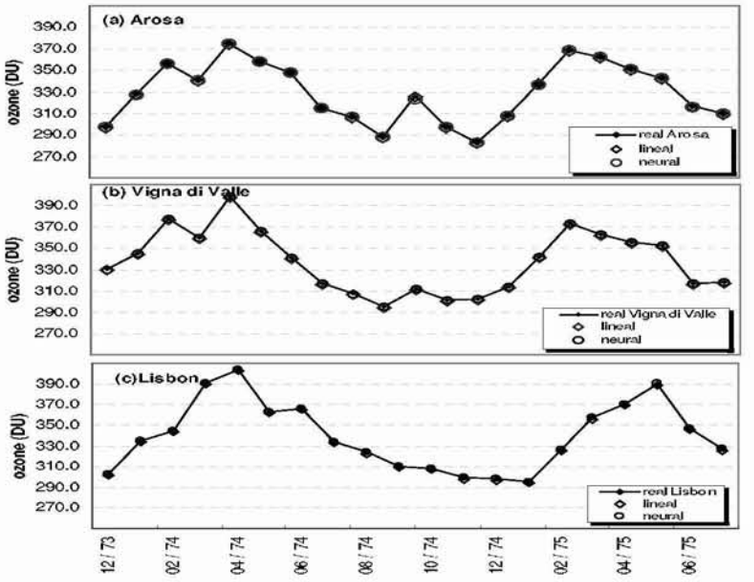


Fig. 1. Interpolation results obtained with $n=2$ for (a) the series of Arosa, (b) Vigna di Valle and (c) Lisbon. Continuous line is for the real series, (◊) stands for the linear model interpolation values, and (○) for the values given by the non linear model

Table 1. Statistics for interpolation results with $n=2$

	Non linear model		*Linear model*	
station	*RMSE*	*VE*	*RMSE*	*VE*
Arosa	2.89	99.94	2.99	99.99
Vigna di Valle	2.64	99.99	3.23	99.99
Lisbon	0.70	99.99	4.46	99.99

3.2 Long Gaps Filling

In the previous point the substitution of isolated missing values has been analysed. Let us consider now an application for filling longer gaps. Neither persistence nor linear interpolation based models are suitable for filling long gaps, because many consecutive values could be missed, even more than a year of measurements. Here below a neural networks based scheme is presented as a promising solution for this kind of problem. The missing data which are going to be filled are those within the winter series of Vigna di Valle. In this case, winter series are made up with the December to March (DJFM) monthly means for every year. The set of predictors must present enough correlation with the predictand variable. During the cold season, the correlation between TO and some atmospheric circulation indices are high enough (R>0.6) to consider those indices as good potential predictors for winter TO. The atmospheric pattern that most influence the considered European areas is the North Atlantic Oscillation (NAO)[2]. The NAO Index (NAOI) informs about the temporal variations of such pattern. The relationship between TO and NAOI for European latitudes has been widely shown in several works [8, 9]. The NAOI monthly time series has been obtained from the Climatic Research Unit (CRU) of the University of East Anglia (UK).

Given the geographic proximity of the stations and the high correlation that Vigna di Valle and Arosa series present, Arosa TO series is also selected as input for the model. The basis of the model used in this case is very simple: by presenting to the net inputs the current value of the series of Arosa and the current value of the time series of the NAOI, the net releases the value of the average TO for the current month at Vigna di Valle. One pyramidal network structure has been used with three input neurons, one hidden layer and one single output cell. The transfer function for the two first layers is the log-sigmoid function, and linear activation has been set for the output. Also here, the Levenberg-Marquardt method is used for training.

The total period of winter data considered for this station is March1957-December2000. We intend to complete the missing values that appear from Dec1980. A moving window has been applied to the set of data employed to train the network, so that to fill the gaps contained within Dec1980 and Dec1985, the Mar1957-Mar1980 data have been used for training. The values given by the model at this stage for the gaps are incorporated to the series, and then the training set is moved five years forward: the March1962-March1985 period is used to train the model in order to obtain the values for the gaps within Dec1985 and Dec1990, and so on. In this way it is easier for the network to track the variations suffered by winter TO throughout the time. In Fig. 2 it is shown the observed real series at Vigna di Valle together with the output series given by the non linear NN model. The correspondence between real and modelled series is quite good. The value of the VE for this model is 70.0 %, i.e. the

[2] The North Atlantic Oscillation (NAO) is the atmospheric circulation pattern which most influences the climate of the North Atlantic Ocean neighbouring lands (from East North America to Western and Central Europe). The NAO determines to a very high extent the wind flow, precipitations and temperatures regimes over such areas.

series given by this non linear model is able to explain 70.0 % of the original series variability. The corresponding RMSE value is of 29.0 %. In this case, with a linear regression model the RMSE and VE values obtained are twice as worse as those obtained with the non linear NNs model.

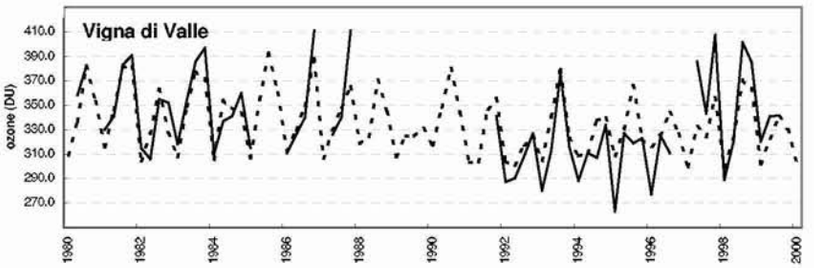


Fig. 2. Observed real series at Vigna di Valle (continuous line and no line for missing intervals) together with the output series given by the non linear NN model (dashed line)

The example shown above is for internal gaps, but this NNs model could also be used to reconstruct past periods of the series for which do not exist measurements for the station of Vigna di Valle, but do exist for the NAO Index and for the station of Arosa. This could be one of the future analysed applications of the NNs models presented here, the reconstruction of data series backwards to pre-instrumental periods. Also the application presented in the next Section is feasible to be used for reconstruction.

4 Future Values Prediction

The study of time series provides valuable information about periodicities and trends in the past which are helpful to predict future behaviours. In this section the estimation of future values for one series is tried by taking known previous values from the same series, or from other ones correlated to the series wanted to predict. This not only allows the forecasting of the considered variable, but also permits to extend the series (forward or backwards) when a longer one is used as predictor.

The longest TO data register is that of the Swiss station of Arosa, where the observations started in the 1920's. The characteristics of this series (see Sect. 3) make it a suitable predictor for Vigna di Valle. Thus, one month ahead prediction is tried for Vigna di Valle by using the n previous values of both Vigna and Arosa, and the current one of the Arosa data register. For the case here analysed the set of data is June 1967–October1980, the period November1975 – October1980 is used for validation.

The model structure used consists of a two layer MLP with k input neurons and one at the output layer. These models ($k+1$ models) have been used before for different

meteorological applications [10]; more complicated structures have been tested for our application without any improvement of the results. For the output layer the linear transfer function has been chosen, the quality of the results seems to be independent from the election of the log-sigmoid or the hyperbolic tangent transfer function for the input layer. As the final model signal, the average of the output of several of these $k+1$ nets is considered. So, to a certain extent we can talk about an ensemble of neural networks, since we are averaging the results given by several $k+1$ structures, with $k=1,..., 2n+1$.

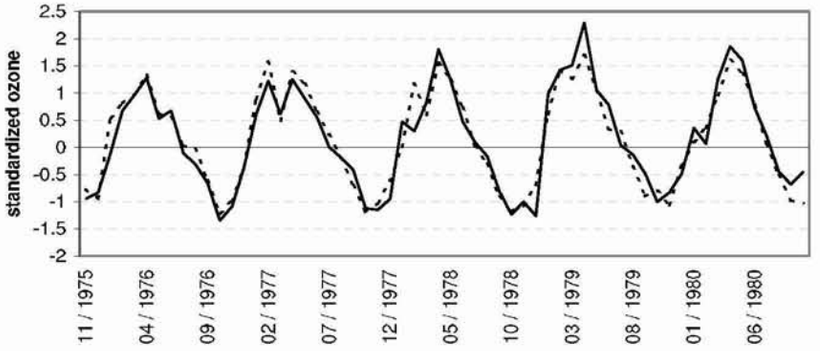


Fig. 3. Prediction results obtained for Vigna di Valle with the NNs ensamble $k+1$ for $n=1$ (---), and the observed real series at the station from November 1975 to October 1980 (—)

The best results are obtained with the ensemble $n=1$, which in spite of having just three neural networks gives out the highest VE and the lowest RMSE. VE values are high (VE > 88.0 %) for all the cases, which means that the shape of the original signal is accurately reproduced by the model. However, RMSE values are greater than 25 % which indicates that some of the original values are over/under estimated. The results of the ensemble $n=1$ together with the observed real series of Vigna di Valle are shown in Fig.3. It can be seen that modelled and original series present high correlation and that this NNs model is able to predict the right sign of the TO variations. So, it is shown that this model, without any modification, is suitable for the forecasting of the sign of TO monthly anomalies.

But if interest is in exact values, the model should be somehow improved in order to get higher precision. The inclusion of a new explanatory variable such as the NAOI for this kind of application is to be studied.

5 Conclusions

In this work different NNs based models have been presented to fill the gaps of incomplete time series of TO, and these non linear models have proved to achieve better results than linear ones. Also a model to predict future values for a TO series has been described. Thus, the high versatility of NNs has allowed us to have a wide variety of applications related to the analysis of TO time series by making quite simple modifications into the model (number of predictors, number of layers and/or neurons), and given the relative simplicity of the proposed structures, further applications based on the models here presented could be developed without the risk of having too complex or time costing schemes.

Acknowledgments

Authors are grateful to the World Ozone and Ultraviolet Radiation Data Centre (WOUDC) for providing ozone data, and to the Climatic Research Unit (CRU) of the University of East Anglia for the NAO Index data series.

References

1. Madronich, S.: UV radiation in the natural and perturbed atmosphere. In: M. Tevini (ed.): UV-B Radiation and Ozone Depletion. A. F. Lewis, New York (1993) 17-69
2. Casale G. R., Meloni, D., Miano, S., Palmieri, S., Siani, A. M., Cappellani, F.: Solar UV-B irradiance and total ozone in Italy: Fluctuations and trends. J. Geophys. Res., 105 (2000) 4895-4901
3. Hansen, B. K.: State of the art of neural networks in meteorology. Mid-term paper for a Neural Networks course at the Technical University of Nova Scotia (1997)
4. Levenberg, K.: A method for the solution of certain problems in least squares. SIAM J. Numer. Anal. 16 (1944) 588-604
5. Marquardt, D.: An algorithm for least squares estimation of nonlinear parameters. SIAM J. Appl. Math. 11 (1963) 431-441
6. Neural Networks Toolbox User's Guide, MatLAB Reference Guide (1997)
7. Widrow, B., Winter, R.: Neural nets for adaptive filtering and adaptive pattern recognition. IEEE Computer, March (1988) 25-39
8. Brönnimann, S., Luterbacher J., Schmutz C., Wanner H., Staehelin J.: Variability of total ozone at Arosa, since 1931 related to atmospheric circulation indices, Geophys. Res. Lett., 27 (2000) 2213-2216
9. Staehelin, J., Mäder, J., Weiss, A. K., Appenzeller, C.: Long-term ozone trends in Northern mid-latitudes with special emphasis on the contribution of changes in dynamics. Physics and Chemistry of the Earth 27 (2002) 461-469
10. Trigo, R.M, Palutikof, J.P.: Simulation of daily temperatures for climate change scenarios over Portugal: a neural network approach. Climate Research, vol. 13 (1999) 45-59

Author Index