

ANNIIP 2007

Kurosh Madani (Ed.)

Artificial Neural Networks and Intelligent Information Processing

Proceedings of the
3rd International Workshop on
Artificial Neural Networks and Intelligent Information
Processing - ANNIIP 2007

INSTICC PRESS



In conjunction with ICINCO 2007
Angers - France, May 2007

Kurosh Madani (Ed.)

Artificial Neural Networks and Intelligent Information Processing

**Proceedings of the
3rd International Workshop on
Artificial Neural Networks and
Intelligent Information Processing
ANNIIPP 2007**

In conjunction with ICINCO 2007
Angers, France, May 2007

INSTICC PRESS
Portugal

Volume Editor

Kurosh Madani
The University of Paris XII
Paris, France

3rd International Workshop on
Artificial Neural Networks and
Intelligent Information Processing
Angers, France, May 2007
Kurosh Madani (Ed.)

Copyright © 2007
INSTICC PRESS
All rights reserved

Printed in Portugal

ISBN: 978-972-8865-86-3
Depósito Legal: 257881/07

Foreword

If much remains still to discover concerning animal brain's internal structure as well as concerning its learning and self-organization mechanisms, I am definitely swayed that bio-inspired Artificial Intelligence has considerably progressed over the last decades. Probably, *as the "progression" of a good wine, attributable to maturity it acquires throughout the years*, part of these progresses is due to the *maturity* of this relatively young area of computational science. However, *"joking aside"*, I am convinced that a number of complementary facts have occupied crucial positions in recent advancements of bio-inspired Artificial Intelligence. Among the most important of them, one could emphasize the nowadays' applicative and technological challenges, emanated from industrial world, opening new dilemmas and decisively highlighted limitations of conventional computational science. Another, but not less important, fact playing a central role in development of bio-inspired artificial systems over the last decade was borders contraction between biological and computational sciences. "Bioinformatics" and "genetics" are significant examples of the numerous benefits issued from the above-mentioned fact. Finally, the third but certainly not the last essential reality is related to the fantastic intellectual dynamics created around Artificial Neural Networks (and more generally around bio-inspired Artificial Intelligence), by an ever-increasing interest of both confirmed and young researchers on this juvenile science generating a reach multidisciplinary synergy between several fields of science representing a wide thematic diversity.

Since 2005, the international workshop on Artificial Neural Networks and Intelligent Information Processing (ANNIIP), within the aforementioned appealing dynamics, and in the frame of the prestigious ICINCO International Conference, aims to offer a privileged occasion to overhaul and exchange the knowledge about further theoretical advances, new experimental discoveries and novel technological improvements in the promising area of the bio-inspired Artificial Intelligence.

Along the lines of the above-expressed "philosophy", born with ANNIIP 2005, aiming to make an inventory of recent advances as well in Artificial Neural Networks as in Intelligent Information Processing areas around a deliberately limited number of presentations, the objective of this book is to convene around a small number of topics a set of relevant papers concerning bio-inspired Artificial Intelligence field. Conformably to our philosophy, the choice of publishing a relatively restricted number

of papers has been motivated on the one hand by the premeditated desire to give a large space to exchanges and discussions, and on the other hand by the strong principle of the presentation of each submitted article by its authors. If “Bio-inspired Artificial Intelligence” and its impact on real-world dilemmas have remained leading themes of this 3rd ANNIIP international workshop’s edition, as in the previous editions, a particular interest has been devoted to the equilibrium between theoretical and applicative aspects.

It is important to remind that scientific relevance and technical quality of a collective volume emerge from quality of its contributors: those who contribute by the high quality of their manuscripts and those who take part in reviewing of submitted papers ensuring the distinction of the book by their valuable expertise. I would like to reedit my acknowledgements to contributors of all accepted papers. Also, I would like to express again my gratitude to Reviewing Board and Program Committee for the valuable work that they accomplished: my heartfelt recognition to those who already were members of ANNIIP Program Committee as well as my sincere thanks to those who kindly accepted to enlarge the Reviewing Board of this new edition of the workshop.

It is also essential to be reminiscent that frequently, creative dynamics is a result of fruitful contacts of humans belonging to a same or to different scientific communities. Again, I would like to express my warm appreciation and my particular gratitude to my friend Prof. Joaquim Filipe, ICINCO 2007 Conference Chair, for his reliance on the young science of “Bio-inspired Artificial Intelligence” and for his confidence on devoting once more this privileged space to the ANNIIP workshop within his valuable conference.

Finally, another key factor in organization of a prestigious conference in so accurate way remains a challenging undertaking requiring a reliable and a solid Organizing Committee. Since its first edition, the ICINCO Organizing Committee has proved its professionalism conformably to this chief point. I would like acknowledge all of the ICINCO Organizing Committee’s members, with a special attention for Marina Carvalho from ICINCO Secretariat who has been particularly a key person in ANNIIP 2007 organization tasks.

Kurosh Madani

PARIS XII – Val de Marne University
France

Workshop Chairs

Kurosh Madani
The University of Paris XII
Paris, France

Program Committee

Veronique Amarger, University of Paris XII - Val de Marne, France
Ezzedine Benbraek, ESSTT, University of Tunis, Tunisia
Abdennasser Chebira, University of Paris XII - Val de Marne, France
Amine Chohra, University of Paris XII - Val de Marne, France
Suash Deb, National Inst. of Science & Technology, India
Colin Frayn, University of Birmingham, UK
Vladimir Golovko, Brest Technical University, Belarus
Jouko Lampinen, Helsinki University of Technology, Finland
Thomas Längle, University of Karlsruhe, Germany
Hichem Maaref, University of Evry - Val d'Essonne, France
David Marroco, CNR di Roma, Italy
Georgy Penev, St-Petersburg Institute Informatics of Russian Academy of Sciences (SPIIRAS), Russia
Anatoly Sachenko, Institute of Intelligent Computer Systems, Ukraine
Rauf Sadikhov, Belarusian State University of Informatics and Radio-electronics, Belarus
Christophe Sabourin, University of Paris XII - Val de Marne, France
Khalid Saeed, Bialystok Technical University, Poland

Table of Contents

Foreword.....	iii
Workshop Chairs	v
Program Committee	v

Papers

A New Model of Associative Memories Network..... <i>Roberto A. Vázquez and Humberto Sossa</i>	3
Forming Neural Networks Design through Evolution..... <i>Eva Volná</i>	13
Architectonics of Thinking: The Conception of Human Brain Organization as Multiprocessing System..... <i>Valery Shyrochin and Vadym Mukhin</i>	21
Initialization by Selection for Multi Library Wavelet Neural Network Training	30
<i>Wajdi Bellil, Mohamed Othmani and Chokri Ben Amar</i>	
ZISC Neural Network Base Indicator for Classification Complexity Estimation	38
<i>Ivan Budnyk, Abdennasser Chebira and Kurosh Madani</i>	
An Improved Architecture for Cooperative and Comparative Neurons (CCNs) in Neural Network.....	48
<i>Md. Kamrul Islam</i>	

Impact of Data Dimensionality Reduction on Neural Based Classification: Application to Industrial Defects.....	56
<i>Matthieu Voiry, Kurosh Madani, Véronique Amarger and Joël Bernier</i>	
Direct and Indirect Classification of High-Frequency LNA Performance using Machine Learning Techniques.....	66
<i>Peter C. Hung, Seán F. McLoone, Magdalena Sánchez, Ronan Farrell and Guoyan Zhang</i>	
Industrial Application Development using Case-based Reasoning...	76
<i>Miroslav Sveda and Ondrej Rysavy</i>	
Noisy Image Processing using the Independent Component Analysis Algorithm AMUSE	83
<i>Salva Nassabay, Ingo R. Keck, Carlos G. Puntonet, Juan M. Górriz, J. Pérez de Inestroaa and Rubén M. Clemente</i>	
Fuzzy Clustering Methods in Multispectral Satellite Image Segmentation	91
<i>Rauf Kh. Sadykbov, Andrey V. Dorogush and Leonid P. Podenok</i>	
Automatic Classification of Spinal Deformity by using Four Symmetrical Features on the Moire Images.....	99
<i>Hyoungseop Kim, Satoshi Nakano, Joo Kooi Tan, Seiji Ishikawa, Yoshinori Otsuka, Hisashi Shimizu and Takashi Shinomiya</i>	
Improved Neural Network-based Face Detection Method using Color Images	107
<i>Yura Kurylyak, Ihor Paliy, Anatoly Sachenko, Kurosh Madani and Amina Chobra</i>	
Artificial Neural Networks Based Approaches: Simulation Toys or Real Solutions?	115
<i>Kurosh Madani</i>	
Author Index	125

PAPERS

A New Model of Associative Memories Network

Roberto A. Vázquez and Humberto Sossa

Centro de Investigación en Computación-IPN
Av. Juan de Dios Batíz, esquina con Miguel Othón de Mendizábal
Mexico City, 07738, Mexico
ravem@ipn.mx, hsossa@cic.ipn.mx

Abstract. An associative memory (AM) is a special kind of neural network that only allows associating an output pattern with an input pattern. However, some problems require associating several output patterns with a unique input pattern. Classical associative and neural models cannot solve this simple task. In this paper we propose a new network composed of several AMs aimed to solve this problem. By using this new model, AMs can be able to associate several output patterns with a unique input pattern. We test the accuracy of the proposal with a database of real images. We split this database of images into four collections of images and then we trained the network of AMs. During training we associate an image of a collection with the rest of the images belonging to the same collection. Once trained the network we expected to recover a collection of images by using as an input pattern any image belonging to the collection.

1 Introduction

An associative memory AM is a special kind of neural network that allows recalling one output pattern given an input pattern as a key that might be altered by some kind of noise (additive, subtractive or mixed). Several models of AMs are described in [1], [2], [3], [5], [9], [10], [11] and [12]. In particular, models described in [5], [9] and [10] cannot handle with mixed noise. Associative model presented in [11] and [12] is robust to mixed noise.

An association between input pattern \mathbf{x} and output pattern \mathbf{y} is denoted as $(\mathbf{x}^k, \mathbf{y}^k)$, where k is the corresponding association. AM \mathbf{W} is represented by a matrix whose component w_{ij} can be seen as the synapse of the neural network. Operator \mathbf{W} is generated from a finite a priori set of know associations, known as the fundamental set of association and is represented as: $\{(\mathbf{x}^k, \mathbf{y}^k) | k = 1, \dots, p\}$ where p is the number of associations. If $\mathbf{x}^k = \mathbf{y}^k \forall k = 1, \dots, p$ then \mathbf{W} is auto-associative, otherwise it is hetero-associative. A distorted version of a pattern \mathbf{x} to be restored will be denoted as $\tilde{\mathbf{x}}$. If an AM \mathbf{W} is fed with a distorted version of \mathbf{x}^k and the output obtained is exactly \mathbf{y}^k , we say that recalling is perfect.

In this paper we present how a network of AMs can be used to recall not just one pattern but several of them given an input pattern. In this proposal an association

between input pattern \mathbf{x} and a collection of output pattern \mathbf{Y} is denoted as, $\{(\mathbf{x}^k, \mathbf{Y}^k) | k=1, \dots, p\}$ where p is the number of association, $\mathbf{Y}^k = \{\mathbf{y}^1, \dots, \mathbf{y}^r\}$ is a collection of output patterns and r is the number of patterns belonging to collection \mathbf{Y} .

The remaining of the paper is organized as follows. In section 2 we describe the associative model used in this research. In section 3 we describe the proposed network of AMs. In section 4 we present the experimental results obtained with the proposal. In section 5 we finally present the conclusions and several directions for further research in this direction.

2 Dynamic Associative Model

The brain is not a huge fixed neural network, as had been previously thought, but a dynamic, changing neural network that adapts continuously to meet the demands of communication and computational needs [8]. This fact suggests that some connections of the brain could change in response to some input stimuli.

Humans, in general, do not have problems to recognize patterns even if these are altered by noise. Several parts of the brain interact together in the process of learning and recalling a pattern. For example, when we read a word the information enters the eye and the word is transformed into electrical impulses. Then electrical signals are passed through the brain to the *visual cortex*, where information about space, orientation, form and color is analyzed. After that, specific information about the patterns passes on the other areas of the *cortex* that integrate visual and auditory information. From here information passes through the *arcuate fasciculus*, a path that connects a large network of interacting brain areas; paths of this pathway connect language areas with other areas involving in cognition, association and meaning, for details see [4] and [7].

Based upon the above example we have defined in our model several interacting areas, one per association we would like the memory to learn. Also we have integrated the capability to adjust synapses in response to an input stimulus.

As we could appreciate from the previous example, before an input pattern is learned or processed by the brain, it is hypothesized that it is transformed and codified by the brain. In our model, this process is simulated using the following procedure recently introduced in [11]:

Procedure 1. Transform the fundamental set of associations into codified patterns and de-codifier patterns:

Input: FS Fundamental set of associations:

{1. Make $d = \text{const}$ and make $(\bar{\mathbf{x}}^1, \bar{\mathbf{y}}^1) = (\mathbf{x}^1, \mathbf{y}^1)$

2. For the remaining couples do {

For $k=2$ to p {

For $i=1$ to n {

$$\bar{x}_i^k = \bar{x}_i^{k-1} + d; \quad \hat{x}_i^k = \bar{x}_i^k - x_i^k; \quad \bar{y}_i^k = \bar{y}_i^{k-1} + d; \quad \hat{y}_i^k = \bar{y}_i^k - y_i^k$$

}} Output: Set of codified and de-codifying patterns.

This procedure allows computing *codified patterns* from input and output patterns denoted by $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$ respectively; $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ are *de-codifying patterns*. Codified and de-codifying patterns are allocated in different interacting areas and d defines of much these areas are separated. On the other hand, d determines the noise supported by our model. In addition a simplified version of \mathbf{x}^k denoted by s_k is obtained as:

$$s_k = s(\mathbf{x}^k) = \mathbf{mid} \mathbf{x}^k \quad (1)$$

where **mid** operator is defined as $\mathbf{mid} \mathbf{x} = x_{(n+1)/2}$.

When the brain is stimulated by an input pattern, some regions of the brain (interacting areas) are stimulated and synapses belonging to those regions are modified.

In our model, we call these regions *active regions* and could be estimated as follows:

$$ar = r(\mathbf{x}) = \arg \left(\min_{i=1}^p |s(\mathbf{x}) - s_i| \right) \quad (2)$$

Once computed the *codified patterns*, the *de-codifying patterns* and s_k we can build the associative memory.

Let $\{(\bar{\mathbf{x}}^k, \bar{\mathbf{y}}^k) | k = 1, \dots, p\}$, $\bar{\mathbf{x}}^k \in \mathbf{R}^n$, $\bar{\mathbf{y}}^k \in \mathbf{R}^m$ a fundamental set of associations (codified patterns). Synapses of associative memory \mathbf{W} are defined as:

$$w_{ij} = \bar{y}_i - \bar{x}_j \quad (3)$$

After computed the *codified patterns*, the *de-codifying patterns*, the reader can easily corroborate that any association can be used to compute the synapses of \mathbf{W} without modifying the results. In short, building of the associative memory can be performed in three stages as:

1. Transform the fundamental set of association into codified and de-codifying patterns by means of previously described Procedure 1.
2. Compute simplified versions of input patterns by using equation 1.
3. Build \mathbf{W} in terms of codified patterns by using equation 3.

2.1 Modifying Synapses of the Associative Model

As we had already mentioned, synapses could change in response to an input stimulus; but which synapses should be modified? For example, a head injury might cause a brain lesion killing hundred of neurons; this entails some synapses to reconnect with others neurons. This reconnection or modification of the synapses might cause that information allocated on brain will be preserved or will be lost, the reader could find more details concerning to this topic in [6] and [13].

This fact suggests there are synapses that can be drastically modified and they do not alter the behavior of the associative memory. In the contrary, there are synapses that only can be slightly modified to do not alter the behavior of the associative memory; we call this set of synapses *the kernel* of the associative memory and it is denoted by \mathbf{K}_w .

In the model we can find two types of synapses: synapses that can be modified and do not alter the behavior of the associative memory; and synapses belonging to the kernel of the associative memory. These last synapses play an important role in recalling patterns altered by some kind of noise.

Let $\mathbf{K}_w \in \mathbf{R}^n$ the kernel of an associative memory \mathbf{W} . A component of vector \mathbf{K}_w is defined as:

$$kw_i = \mathbf{mid}(w_{ij}), j = 1, \dots, m \quad (4)$$

According to the original idea of our proposal, synapses that belong to \mathbf{K}_w are modified as a response to an input stimulus. Input patterns stimulate some *active regions*, interact with these regions and then, according to those interactions, the corresponding synapses are modified. Synapses belonging to \mathbf{K}_w are modified according to the stimulus generated by the input pattern. This adjusting factor is denoted by Δw and can be computed as:

$$\Delta w = \Delta(\mathbf{x}) = s(\bar{\mathbf{x}}^{ar}) - s(\mathbf{x}) \quad (5)$$

where *ar* is the index of the *active region*.

Finally, synapses belonging to \mathbf{K}_w are modified as:

$$\mathbf{K}_w = \mathbf{K}_w \oplus (\Delta w - \Delta w_{old}) \quad (6)$$

where operator \oplus is defined as $\mathbf{x} \oplus e = x_i + e \quad \forall i = 1, \dots, m$. As you can appreciate, modification of \mathbf{K}_w in equation 6 depends of the previous value of Δw denoted by Δw_{old} obtained with the previous input pattern. Once trained the **AM**, when it is used by first time, the value of Δw_{old} is set to zero.

2.2 Recalling a Pattern using the Proposed Model

Once synapses of the associative memory have been modified in response to an input pattern, every component of vector $\bar{\mathbf{y}}$ can be recalled by using its corresponding input vector $\bar{\mathbf{x}}$ as:

$$\bar{y}_i = \mathbf{mid}(w_{ij} + \bar{x}_j), j = 1, \dots, n \quad (7)$$

In short, pattern $\bar{\mathbf{y}}$ can be recalled by using its corresponding key vector $\bar{\mathbf{x}}$ or $\tilde{\mathbf{x}}$ in six stages as follows:

1. Obtain index of the active region *ar* by using equation 2.
2. Transform \mathbf{x}^k using de-codifying pattern $\hat{\mathbf{x}}^{ar}$ by applying the following transformation: $\hat{\mathbf{x}}^k = \mathbf{x}^k + \hat{\mathbf{x}}^{ar}$.
3. Compute adjust factor $\Delta w = \Delta(\hat{\mathbf{x}})$ by using equation 5.
4. Modify synapses of associative memory \mathbf{W} that belong to \mathbf{K}_w by using equation 6.
5. Recall pattern $\hat{\mathbf{y}}^k$ by using equation 7.

6. Obtain \mathbf{y}^k by transforming $\widehat{\mathbf{y}}^k$ using de-codifying pattern $\widehat{\mathbf{y}}^{ar}$ by applying transformation: $\mathbf{y}^k = \widehat{\mathbf{y}}^k - \widehat{\mathbf{y}}^{ar}$.

The formal set of prepositions that support the correct functioning of this dynamic model can be found in [14].

3 Architecture of the Network

Classical AMs (see for example [1], [2], [3], [5], [9], [10], [11] and [12]) are able to recover a pattern (an image) from a noisy version of it. In their original form classical AMs are not useful when image is altered by image transformations, such as translations, rotations, and so on.

The network of AMs proposed in this paper is robust under some of these transformations. Taking advantage of this fact, we can associate different versions of an image (rotated, translated and deformed) to an image.

Our task is to propose a network of AMs aimed to associate an image with other images belonging to the same collection. In order to achieve this, first suppose we want to associate images belonging to a collection with an image of the same collection using an AM. A good solution could be to compute the average image of whole images belonging to the collection and then associate the average image with any image that belongs to the collection. The same solution can be applied to other collections. Once computed the average images from different collections and chosen the images to be associated, we can train the AM as was described in section 2.

Until this point the AM only can recover an association between a collection of input patterns \mathbf{X} and output pattern \mathbf{y} denoted as, $\{(\mathbf{X}^k, \mathbf{y}^k) | k = 1, \dots, p\}$ where p is the number of association, $\mathbf{X}^k = \{\mathbf{x}^1, \dots, \mathbf{x}^r\}$ is a collection of input patterns and r is the number of patterns belonging to collection \mathbf{X} . This means that it can only be recovered the associated image using any image from a collection. However, we would like to get the inverse result; instead of recovering the associated image using any image from a collection, we would like to recover all the images belonging to the collection using any image of the collection.

To achieve this goal we will train a network of AMs built as in previous sections. Each AM will associate all the images of a collection with one image of this collection. This implies that for recovering all images of a given collection, we would need r AMs, where r is the number of images belonging to the collection. The network architecture of AMs needed for recovering a collection of images is shown in Fig. 1.

In order to train the network of r AMs, first of all we need to know the number of collections we want to recover. Training phase is done as follows:

1. Transform each image into a vector.
2. Build n collections of images $[\mathbf{C}^n]_{q \times r}$ where q is the number of pixels of each image and r the number of images.

3. Let $[\mathbf{AI}]_{q \times n}$ a matrix of average images. For $k=1$ to n compute the average image as:

$$\mathbf{AI}_k = \frac{\sum_{s=1}^r \mathbf{CI}_s^k}{r} \quad (8)$$

4. For $s=1$ to r build an \mathbf{AM}^s as described in Section 2.2. For $k=1$ to n $\mathbf{x}^k = \mathbf{AI}_k$ and $\mathbf{y}^k = \mathbf{CI}_s^k$

Once trained the network of AMs, when is fed with any image of a collection, each AM will respond with an image that belongs to the collection. To recover a collection of images we just operate each AM with the input image as described in Section 2.3.

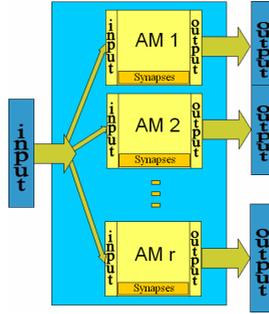


Fig. 1. Architecture of a network of AMs for recalling a collection of patters using an input pattern.

4 Experimental Results

In this section the accuracy of the proposal is tested using different collections of images shown in Fig. 2.

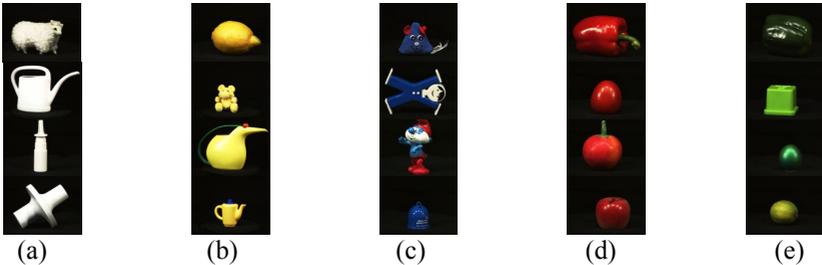


Fig. 2. (a-e) Collections of images taken from the Amsterdam Library of Objects Images (ALOI).

Twenty complex images were grouped into five collections composed by four images, see Fig. 2. After that, we proceeded to train the network of AMs as was

explained in Section 3. For this problem our network is composed of four AMs. It is important to say that the number of images composing a collection could be any, the only restrictions to guaranty perfect and robust recall is that patterns (images) satisfy propositions described in [14].

Once trained the network, four experiments were performed to test the accuracy of the proposal. The first experiment consisted on recovering a collection of images using any image of the collection in order to verify how much robust is the proposal under image deformations. Second experiment consisted on recovering a collection of images using any image of the collection altered by mixed noise in order to verify how much robust is the proposal under deformations and noisy version of the images. In the third experiment each image of the training set was rotated (from 0 to 360). We then used them to fed the network of AMs in order to verify how much robust is the proposal under deformations and rotations. Finally for forth experiment the images of the training set were rotated (from 0 to 360) and translated, and then used them to fed the network of AMs in order to verify how much robust is the proposal under deformations, rotations and translations.

The accuracy of the proposal was of 100% in the first experiment. The five collections of images where perfectly recovered by using any image of the collection (20 images), in Table 1 are shown some results obtained in this experiment. Remember that we train the network of AMs with average images, so then; when we fed the network with an image of any collection this image could be seen as a deformed version of the average images. The results provided by our proposal in this experiment show that the associative model used to train the network of AMs is robust under deformations. Something important to say is that if we use other associative models for training the network of AMs such as morphological or median AMs, the collections might not be correctly recovered due to they are not robust under these kind of transformations or deformations.

Table 1. Some results obtained for the first experiment. As you can appreciate all sets of images were perfectly recovered.

Input image	Image recovered by each AM.							
								
								
								
								
								

The accuracy of the proposal, in the second experiment, was of 100%. The five collections of images where recovered perfectly by using any image of the collection even when this images were altered by noise (200 images). As you can see in Table 2, despite of the level of noise add to the images, the collections were correctly

recovered. Despite of other associative models are robust to this kind of noise, they might not recover the all collections due to they are only robust under additive, subtractive and mixed noise but not to deformations.

Table 2. Some results obtained for the second experiment. Despite of the noise added to the images, all sets of images were again correctly recovered.

Input image	Image recovered by each AM							
								
								
								
								
								

The accuracy of the proposal, in the third experiment, was also of 100%. The five collections of images where recovered perfectly even when rotated version of the images were used (700 images), see in Table 3. Some important to say is, to our knowledge, neither morphological AMs nor other classical models are robust under rotations. Due to we used simplified patterns using **mid** operator and due to this operator is invariant to rotations the accuracy of the proposal was of 100%.

Table 3. Some results obtained for the third experiment. Despite of the noise added to the images and rotations, all sets of images were again correctly recovered.

Input image	Image recovered by each AM							
								
								
								
								
								

Finally, the accuracy of the proposal, in the fourth experiment was of 40%. In this experiment with 700 images; with some images we recall a collection, but with some other images (when patterns do not satisfied the proposition which guarantee robust recall) the collections were not recalled, see Table 4. However, the results obtained by our proposal are acceptable if they are compared with the results provided by order associative models (less of 10% of accuracy).

Table 4. Some results obtained for the forth experiment. With some images we recall a collection, but with some other images (when patters do not satisfied the proposition which guarantee robust recall) the collections was not recalled.

Input image		Image recovered by each AM							
									
									
									
									
									

In general, the accuracy of the proposal with different banks of images (altered by noisy and rotated) was of 100%. This was due to the input patterns (the images) satisfy the propositions presented in [14]. If these patterns do not satisfy these propositions, as images used in experiment four (translated and rotated images), the accuracy of the proposal diminish. However, the results provided by our proposal up-performed the results provide by other associative models.

5 Conclusions and Directions for Further Research

In this paper we have proposed a network of AMs. This network is useful for recalling a collection of output patterns using an input pattern as a key. The network is composed by several dynamic associative memories (DAM). This DAM is inspired in some aspects of human brain. The model, due to plasticity of its synapses and functioning, is robust under some transformations as rotation, translation and deformations.

In addition, we describe an algorithm for training a network of AMs codifying the images of a collection by using an average image. Once computed the average images we proceed to training the network of AMs.

The network is capable to recall a collection of images (patterns) even if images are altered by noise or suffer some deformations, rotations and translations.

Through several experiments we have shown the efficiency of the proposal. In the first three experiments the proposal provided an accuracy of 100%. Even when the images were altered with mixed noise and rotated, the network of AMs recovered the corresponding collection. When object in images suffer translations, the accuracy of the proposal diminish. This is because most of the patterns under this transformation do not satisfied the propositions that guarantee robust recall. However, the results provided by our proposal up-performed the results provide by other associative models.

The performed experiments could be seen as an application for image retrieval problems. We could say that we have developed a small system able to recover a

collection of images (previously organized), even in the presence of altered versions of the images.

Nowadays we are working and directed this research to solve real problems in images retrieval system. We are focusing our efforts to propose new associative models able to associate and recall images under more complex transformations. Furthermore, this new models have to work with images of much more complicated objects such as flowers, animals, cars, faces, etc.

Acknowledgements

This work was economically supported by COTEPABE-IPN, SIP-IPN under grant 20071438 and CONACYT under grant 46805.

References

1. K. Steinbuch. 1961. Die Lernmatrix. *Kybernetik*, 1(1):26-45.
2. J. A. Anderson. 1972. A simple neural network generating an interactive memory. *Mathematical Biosciences*, 14:197-220.
3. J. J. Hopfield. 1982. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79: 2554-2558.
4. M. Kutas and S. A. Hillyard. 1984. Brain potentials during reading reflect word expectancy and semantic association. *Nature*, 307:161-163.
5. G. X. Ritter et al. 1998. Morphological associative memories. *IEEE Transactions on Neural Networks*, 9:281-293.
6. Iva Reinvan. 1998. Amnesic disorders and their role in cognitive theory. *Scandinavian Journal of Psychology*, 39(3): 141-143.
7. C. J. PRICE. 2000. The anatomy of language: contributions from functional neuroimaging. *Journal of Anatomy*, 197(3): 335-359.
8. S. B. Laughlin and T. J. Sejnowski. 2003. Communication in neuronal networks. *Science*, 301:1870-1874.
9. P. Sussner. 2003. Generalizing operations of binary auto-associative morphological memories using fuzzy set theory. *Journal of mathematical Imaging and Vision*, 19(2):81-93.
10. G. X. Ritter, G. Urcid, L. Iancu. 2003. Reconstruction of patterns from noisy inputs using morphological associative memories. *Journal of mathematical Imaging and Vision*, 19(2):95-111.
11. H. Sossa, R. Barrón, R. A. Vázquez. 2004. Transforming Fundamental set of Patterns to a Canonical Form to Improve Pattern Recall. *LNAI*, 3315:687-696.
12. H. Sossa, R. Barrón, R. A. Vázquez. 2004. New associative memories for recall real-valued patterns. *LNCS*, 3287:195-202.
13. K. D. Jovanova-Nesic and B. D. Jankovic. 2005. The Neuronal and Immune Memory Systems as Supervisors of Neural Plasticity and Aging of the Brain: From Phenomenology to Coding of Information. *Annals of the New York Academy of Sciences*, 1057: 279-295
14. R. A. Vázquez and H. Sossa. 2007. A new associative memory with dynamical synapses. Submitted to *IJCNN* 2007.

Forming Neural Networks Design through Evolution

Eva Volná

University of Ostrava, 30ht dubna st. 22, 701 03 Ostrava, Czech Republic
eva.volna@osu.cz

Abstract. Neuroevolution techniques have been successful in many sequential decision tasks such as robot control and game playing. This paper aims at evolution in artificial neural networks (e.g. neuroevolution). Here is presented a neuroevolution system evolving populations of neurons that are combined to form the fully connected multilayer feedforward network with fixed architecture. In this paper, the transfer function has been shown to be an important part of architecture of the artificial neural network and have significant impact on an artificial neural network's performance. In order to test the efficiency of described method, we applied it to the alphabet coding problem.

1 Introduction to Neuroevolution

Evolutionary algorithms refer to a class of population-based stochastic search algorithms that are developed from ideas and principles of natural evolution. They include [1] evolution strategies, evolutionary programming, and genetic algorithms. Evolutionary algorithms are particularly useful for dealing with large complex problems which generate many local optima. They are less likely to be trapped in local minima than traditional gradient-based search algorithms. They do not depend on gradient information and thus are quite suitable for problems where such information is unavailable or very costly to obtain or estimate. They can even deal with problems, where no explicit and/or exact objective function is available. These features make them much more robust than many other search algorithms. Fogel [2] and Bäck et al. [3] give a good introduction to various evolutionary algorithms for optimization. One important feature of all these algorithms is their population-based search strategy. Individuals in a population compete and exchange information with each other in order to perform certain tasks. A general framework of evolutionary algorithms can be described as follows:

```
generate the initial population G(0) at random, set i=0
REPEAT
  ..Evaluate each individual in the population;
  ..Select parents from G(i) based on their fitness
    in G(i);
  ..Apply search operators to parents and produce
    offspring which form G(i+1);
  ..i = i+ 1;
UNTIL 'termination criterion' is satisfied
```

Neuroevolution represents a combination of neural networks and evolutionary algorithms, where neural networks are the phenotype being evaluated. The genotype is a compact representation that can be translated into an artificial neural network. Evolution has been introduced into artificial neural networks at roughly three different levels: connection weights, architectures, and learning rules. The evolution of connection weights provides a global approach to connection weights training, especially when gradient information of the error function is difficult or costly to obtain. Due to the simplicity and generality of the evolution and the fact that gradient-based training algorithms often have to be run multiple times in order to avoid being trapped in a poor local optimum, the evolutionary approach is quite competitive. The evolution of architectures enables artificial neural networks to adapt their topologies to different tasks without human intervention and thus provides an approach to automatic artificial neural network design. Simultaneous evolution of artificial neural network architectures and connection weights generally produces better results. The evolution of learning rules in artificial neural networks can be used to allow an artificial neural network to adapt its learning rule to its environment. In a sense, the evolution provides artificial neural network with the ability of learning to learn. Global search procedures such as evolutionary algorithms are usually computationally expensive. It would be better not to employ evolutionary algorithms at all three levels of evolution in neural networks. It is, however, beneficial to introduce global search at some levels of evolution, especially when there is little prior knowledge available at that level and the performance of the artificial neural network is required to be high, because the trial-and-error or heuristic methods are very ineffective in such circumstances. With the increasing power of parallel computers, the evolution of large artificial neural networks becomes feasible. Not only can such evolution discover possible new artificial neural network architectures and learning rules, but it also offers a way to model the creative process as a result of artificial neural network's adaptation to a dynamic environment.

2 Overview of the Evolution of Node Transfer Functions

The discussion on the evolution of architectures so far only deals with the topological structure of architecture. The transfer function of each node in the architecture has been usually assumed that is fixed and predefined by human experts, at least for all the nodes in the same layer. Little work has been only done on the evolution of node transfer function up to now. Mani proposed a modified backpropagation, which performs gradient descent search in the weight space as well as the transfer function space [4], but connectivity of artificial neural networks was fixed. Lovel and Tsoi investigated the performance of Neocognitrons with various S-cell and C-cell transfer functions, but did not adopt any adaptive procedure to search for an optimal transfer function automatically [5]. Stork et al. [6] were, to our best knowledge, the first to apply evolutionary algorithms to the evolution of both topological structures and node transfer functions even though only simple artificial neural networks with seven nodes were considered. The transfer function was specified in the structural genes in their genotypic representation. It was much more complex than the usual sigmoid function because authors in [6] tried to model biological neurons. White and Ligomenides [7]

adopted a simpler approach to the evolution of both topological structures and node transfer functions. For each individual (i.e. artificial neural network) in the initial population, 80% nodes in the artificial neural network used the sigmoid transfer function and 20% nodes used the Gaussian transfer function. The evolution was used to decide the optimal mixture between these two transfer functions automatically. The sigmoid and Gaussian transfer function themselves were not evolvable. No parameters of the two functions were evolved. Liu and Yao [1] used evolutionary programming to evolve artificial neural networks with both sigmoidal and Gaussian nodes. Rather than fixing the total number of nodes and evolve mixture of different nodes, their algorithm allowed growth and shrinking of the whole artificial neural network by adding or deleting a node (either sigmoidal or Gaussian). The type of node added or deleted was determined at random. Hwang et al. [8] went one step further. They evolved topology of artificial neural network, node transfer function, as well as connection weights for projection neural networks. Sebald and Chellapilla [9] used the evolution of node transfer function as an example to show the importance of evolving representations. Representation and search are the two key issues in problem solving. Co-evolving solutions and their representations may be an effective way to tackle some difficult problems where little human expertise is available. In principle, the difference in transfer functions could be as large as that in the function type, e.g. that between a hard limiting threshold function and Gaussian function, or as small as that in one of parameters of the same type of function, e.g. the slope parameter of the sigmoid function. The decision on how to encode transfer functions in chromosomes depends on how much prior knowledge and computation time is available. This suggests some kind of indirect encoding method, which lets developmental rules to specify function parameters if the function type can be obtained through evolution, so that more compact chromosomal encoding and faster evolution can be achieved. One point worth mentioning here is the evolution of both connectivity and transfer functions at the same time [6] since they constitute a complete architecture. Encoding connectivity and transfer functions into the same chromosome makes it easier to explore nonlinear relations between them. Many techniques used in encoding and evolving connectivity could equally be used here.

3 Evolution Design of Neural Networks With Fixed Topology

In the paper, the transfer function has been shown to be an important part of architecture of the artificial neural network, one has significant impact on artificial neural network's performance. Here is presented a neuroevolution system evolving populations of neurons that are combined to form the fully connected multilayer feedforward network with fixed architecture. Neuroevolution evolves transfer functions of each unit in hidden and output layers of the network. The system maintains diversity in the population, because a dominant neural phenotype is likely to end up in the same network more than once. As several different types of neurons are usually necessary to solve a problem, networks with too many copies of the same neuron are likely to fail. The dominant phenotype then loses fitness and becomes less dominant. The system works well because it makes sure neurons get the credit they deserve, unlike some other neuroevolution techniques, where bad neurons can share

in a good network or good neurons can be brought down by their network. It also works by decomposing the task, breaking the search into smaller, more manageable parts.

In the following is described a method of automatic searching the node transfer function architecture in multilayer feedforward network: First, we must propose neural network architecture before the main calculation. We get the number of input (m) and output (o) units from the training set. Next, we have to define the number of hidden units (h) that is very confounding issue, because it is generally more difficult to optimize large networks than small ones. Thereafter the process of evolutionary algorithms is applied. Chromosomes are generated for every individual from the initial population as follows, see Fig. 1:

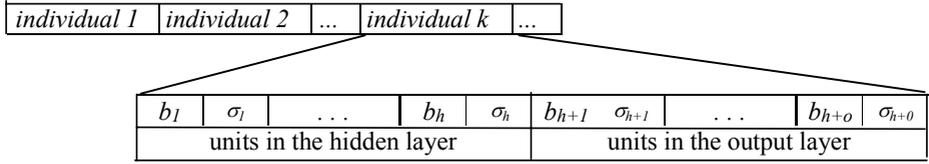


Fig. 1. Population of individuals and their chromosomes.

Symbols b_i ($i = 1, \dots, h+o$) refers to varies types of activation functions [10]:

- $b_i = 1$, if the activation function is a *binary sigmoid function*:

$$f(x) = \frac{1}{1 + \exp(-\sigma x)}. \quad (1)$$

where σ is the steepness parameter, which value is set in the initial population randomly (e.g. $\sigma_i \in (0; 7>)$).

- $b_i = 2$, if the activation function is a *binary step function with threshold θ* .

$$f(x) = \begin{cases} 1 & \text{if } x \geq \theta \\ 0 & \text{if } x < \theta \end{cases}. \quad (2)$$

the steepness parameter σ_i is not define here thus we assigned value 0 to it.

- $b_i = 3$ if the activation function is a *Gaussian function*:

$$f(x) = \exp(-x^2). \quad (3)$$

the steepness parameter σ_i is not define here thus we assigned value 0 to it.

- $b_i = 4$, if the activation function is a *saturated linear function*:

$$f(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } 0 \leq x \leq 1. \\ 1 & \text{if } x > 1 \end{cases} \quad (4)$$

the steepness parameter σ_i is not define here thus we assigned value 0 to it.

Next, we calculate an error value (E) between the desired and the real output after defined partial training with genetic algorithms. Adaptation of each individual starts with randomly generated weight values that are the same for each neural network in the given population. On the basis of it is calculated a fitness function for every individual as follows:

$$Fitness_i = E_{max} - E_i. \quad (5)$$

for $i = 1, \dots, N$,

where E_i is error for the i -th network after a partial adaptation;

E_{max} is a maximal error for the given task,

$E_{max} = o \times pattern$

(o is number of output units and $pattern$ is number of patterns);

N is the number of individuals in the population.

All of the calculated fitness function values of the two consecutive generations are sorted descending and the neural network representation attached to the first half creates the new generation. For each fitness function is calculated the probability of reproduction its existing individual by standard method [11]. One-point *crossover* was used to generate two offspring. If the input condition of *mutation* is fulfilled (e.g. if a randomly number is generated, that is equal to the defined constant), one of the individual is randomly chosen. There is randomly replaced one place in its genetic representation by a random value from the set of permitted values. Our calculation finishes, when the population is composed only from the same individuals.

4 Experiments

In order to test the efficiency of described method, we applied it to the alphabet coding problem that exists in cryptography. Neural networks can be also used in encryption or decryption algorithms, where parameters of adapted neural networks are included to cipher keys [12]. Cipher keys must have several heavy attributes. The best one is the singularity of encryption and cryptanalysis [13]. Encryption is a process in which we transform the open text (e.g. news) to cipher text according to rules. Cryptanalysis of the news is the inverse process, in which the receiver of the cipher transforms it to the original text. The open text is composed from alphabet characters, digits and punctuation marks. The cipher text has usually the same composition as the open text.

We worked with multilayer neural networks, which topologies were based on the training set (see Table 1). The chain of chars of the plain text in a training set is equivalent to a binary value that is 96 less than its ASCII code. The cipher text is then a random chain of bits.

Table 1. The set of patterns (the training set).

THE PLAIN TEXT			THE CIPHER TEXT	THE PLAIN TEXT			THE CIPHER TEXT
Char	ASCII code (DEC)	The chain of bits	The chain of bits	Char	ASCII code (DEC)	The chain of bits	The chain of bits
a	97	00001	000010	n	110	01110	011100
b	98	00010	100110	o	111	01111	101000
c	99	00011	001011	p	112	10000	001010
d	100	00100	011010	q	113	10001	010011
e	101	00101	100000	r	114	10010	010111
f	102	00110	001110	s	115	10011	100111
g	103	00111	100101	t	116	10100	001111
h	104	01000	010010	u	117	10101	010100
i	105	01001	001000	v	118	10110	001100
j	106	01010	011110	w	119	10111	100100
k	107	01011	001001	x	120	11000	011011
l	108	01100	010110	y	121	11001	010001
m	109	01101	011000	z	122	11010	001101

The initial population contains 30 three-layer feedforward neural networks. Each network architecture is 5 - 5 - 6 (e.g. five units in the input layer, five units in the hidden layer, and six units in the output layer), because the alphabet coding problem is not linearly separable and therefore we cannot use neural network without hidden units. The nets are fully connected. We use the genetic algorithm with the following parameters: probability of mutation is 0.01 and probability of crossover is 0.5. Adaptation of each neural network in given population starts with randomly generated weight values that are the same for each neural network in the population. We also used genetic algorithms with the same parameters for the partial neural network adaptation, where number of generations for a partial adaptation was 500. Its chromosome representation is described in [14].

History of the error function is shown in the figure 3. There are shown average values of error functions in the given population. Other numerical simulations give very similar results. The “*binary sigmoid function*” represents an average value after adaptation with the binary sigmoid activation function consecutively with all

steepness parameters $\sigma = \{1,2,3,4,5,6,7\}$. The “*binary step function*” represents an adaptation with the binary step activation function (with the threshold θ), the “*saturated linear function*” represents an adaptation with the saturated linear activation function, and “*Gaussian activation*” represents an adaptation with the Gaussian activation function. Each of these mentioned representations is associated with all units in given neural network architecture. Opposite of this, the “*best individual*” represents an adaptation of the best individual in population, which chromosome is the following, see Fig. 2:

b_1	σ_1	b_2	σ_2	b_3	σ_3	b_4	σ_4	b_5	σ_5	b_6	σ_6	b_7	σ_7	b_8	σ_8	b_9	σ_9	b_{10}	σ_{10}	b_{11}	σ_{11}
1	5	1	7	1	1	3	0	1	5	3	0	1	7	3	0	1	5	1	6	1	2
units in the hidden layer												units in the output layer									

Fig. 2. The “best individual” chromosome in the last population.

5 Conclusions

All networks solve the alphabet coding task in our experiment, but artificial neural network with evolving transfer functions of each unit works well, because several different types of neurons are usually necessary to solve a problem. We can see that the proposed technique is really efficient for the presented purpose, see the Fig. 3. Networks with too many copies of the same neuron work usually worse.

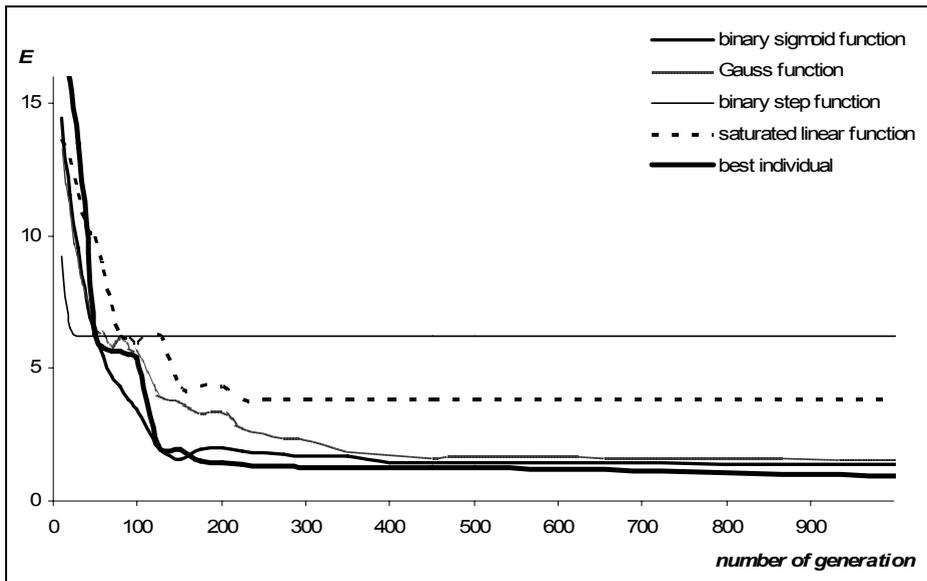


Fig. 3. The error function history.

Here, the transfer function is shown to be an important part of architecture of the artificial neural network and have significant impact on artificial neural network’s

performance. Transfer functions of different units can be different and decided automatically by an evolutionary process, instead of assigned by human experts. In general, nodes within a group, like layer, in an artificial neural network tend to have the same type of transfer function with possible difference in some parameters, while different groups of nodes might have different types of transfer function.

References

1. Liu, Y. and Yao, X. "Evolutionary design of artificial neural networks with different nodes". In *Proc. 1996 IEEE Int. Conf. Evolutionary Computation (ICEC'96)*, Nagoya, Japan, pp. 670–675.
2. Fogel, D. B., *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. New York: IEEE Press, 1995.
3. Bäck, T. Hammel, U. and Schwefel, H.-P. "Evolutionary computation: Comments on the history and current state," *IEEE Trans. Evolutionary Computation*, vol. 1, pp. 3–17, Apr. 1997.
4. Mani, G. "Learning by gradient descent in function space". In *Proc. IEEE Int. Conf. System, Man, and Cybernetics*, Los Angeles, CA, 1990, pp. 242–247.
5. Lovell D. R. and Tsoi, A. C. *The Performance of the Neocognitron with Various S-Cell and C-Cell Transfer Functions*, Intell. Machines Lab., Dep. Elect. Eng., Univ. Queensland, Tech. Rep., Apr. 1992.
6. Stork, D. G. Walker, S. Burns, M. and Jackson, B. "Preadaptation in neural circuits". In *Proc. Int. Joint Conf. Neural Networks*, vol. I, Washington, DC, 1990, pp. 202–205.
7. White D. and Ligomenides, P. "GANNet: A genetic algorithm for optimizing topology and weights in neural network design". In *Proc. Int. Workshop Artificial Neural Networks (IWANN'93)*, Lecture Notes in Computer Science, vol. 686. Berlin, Germany: Springer-Verlag, 1993, pp. 322–327.
8. Hwang, M. W. Choi, J. Y. and Park, J. "Evolutionary projection neural networks". In *Proc. 1997 IEEE Int. Conf. Evolutionary Computation, ICEC'97*, pp. 667–671.
9. Sebald, A. V. and Chellapilla, K. "On making problems evolutionarily friendly, part I: Evolving the most convenient representations". In Porto, V. W., Saravanan, N., Waagen, D. and Eiben, A. E. (Eds.): *Evolutionary Programming VII: Proc. 7th Annu Conf. Evolutionary Programming*, vol. 1447 of Lecture Notes in Computer Science, Berlin, Germany: Springer-Verlag, 1998, pp. 271–280.
10. Volná E. "Evolution design of an artificial neural network with fixed topology", In R. Matoušek, P. Ošmera (eds.): *Proceedings of the 12th International Conference on Soft Computing, Mendel'06*, Brno, Czech Republic, 2006, pp. 1-6.
11. Lawrence, D. *Handbook of genetic algorithms*. Van Nostrand Reinhold, New York 1991.
12. Volná, E. „Using Neural network in cryptography“. In P. Sinčák, J. Vaščák, V. Kvasnička, R. Mesiár (eds.): *The State of the Art in Computational Intelligence*. Physica-Verlag Heidelberg 2000. pp.262-267.
13. Garfínger, S. *PGP: Pretty Good Privanci*. Computer Press, Praha 1998.
14. Volná, E. „Learning algorithm which learns both architectures and weights of feedforward neural networks“. *Neural Network World. Int. Journal on Neural & Mass-Parallel Comp. and Inf. Systems*. 8 (6): 653-664, 1998.

Architectonics of Thinking: The Conception of Human Brain Organization as Multiprocessing System

Valery Shyrochin and Vadym Mukhin

National Technical University of Ukraine “Kiev Polytechnic Institute”
03056 Kiev, Ukraine
{vpsh514, mukhin}@comsys.ntu-kpi.kiev.ua

Abstract. This paper is devoted to the development of hypothesis about human brain “cartation”, i.e. spatially distributed structure of human brain systems for cogitative activities maintenance, containing neural parts with various degree of rigidity and different functional tasks (cartoids), which was suggested by Institute of Brain of the Russian Academy of Sciences (St.-Petersburg) under supervision of acad. N. Bekhtereva.

In the paper is suggested the conception of the structurally-logic organization of thinking processes, which defines the main attitudes between the emotional-strong-willed (unconscious) and the intellectual-ethical (conscious) components (cartoids-processors) of thinking for the situation analysis and for the choice of actions ways by the person during decisions making and realization. The conception is based on the architectonics of the artificial intellect of the future generations for creation of the emotionally and morally-oriented knowledge bases and supercomputers, capable to realize all the processes of the productive creative thinking.

1 Introduction

Nowadays, the important scientific problem is the development of computer systems with the artificial intellect, which facilities are more and more closer to the human abilities to think productive. [1] The artificial intellect systems should in the shortest terms make, instead of the person, the decision, which are, for example, by validity and speed parameters, better than decisions, that are making by the person. Due to wide-spread using of the computer systems and the global and corporate networks, consisting from the personal and professional computers, the interest to the playing, planning and other systems for decisions making, has essentially increased. [2],[3] The meta-mechanisms, that simulating the functions of human intellect, are the base components of the intellectual software, allowing automate the cogitative activities not only for the ordinary computer users, but also for the programmers.

Thus, the actual problem is researching of the architectonics (meta-models and neural processors) of human thinking, which will allow to design the structurally-logic organization of artificial neural intellect, which is capable to realize both conscious formal-logic, and unconscious emotional-intuitive mechanisms of the creative thinking and the informal decisions making by person. [1]

2 The Architectonics of Common Bus-based Images Processing with Neural Networks

The important factor of human's reactions is the verbal expression of the person's attitude to a perceived image on the basis of perceived before actions ways and of the management skills (articulation) of the speech mechanism for the ideas expression. The attitude to *Wi*-concept can be reflected in speech, i.e. can be determined the word or the semantically connected chain of words that are formed by a neural-stack of a actions ways window (articulation windows). The corresponded neural networks with their final neurons are directly connected to reflex arches of the speech mechanism muscles management: a mouth, tongue, throats and breath which realize "knitting" of sound speech images structures, which probably reflecting the real World and allowing coordinate the actions of the people's groups and collectives. The general structure of effectors networks connections of neural-stacks for various windows of the neural-screen of human brain bark is shown on Fig. 1.

Using the effectors networks of brain organization as example there is an opportunity precisely to show differences of databases, for example, for recognition visual both sound images, and bases of knowledge which collect in the actions ways window for realization of corresponding skills and actions algorithms, including the ways and algorithms of a "convincing articulation" which can be perceived as logic thinking.

In databases for recognition of visual or sound images are accumulated and "passively" stored the big enough volumes of "etalon" images, including phonetic sound and sign visual images of words and word-combinations which provide recognition with the general perception only as: "has understood" or "has learned", so it is possible to generate corresponding, associated with this concept, actions.

To understand are means to save up some set of "reference" sound or visual images which will allow for neural-stacks to distinguish images similar to them and to generate corresponding signals *Wi*-"known", which can have both verbal and emotional form: "has understood!"

To know are means to save up some set of ways for "correct" articulation which will allow to connect "logically" and "convincingly" to express in the sounded speech images some concepts and their correct or desirable attitudes to concrete discussed problem area.

The skills of "convincing" speech are got during all human life and, first of all, during dialogue, education and training on more than fifteen-year of base and high educational schools and universities speech trainings and competitions. Knowledge allow to generate the "adequate", clear for surrounding humans reaction as corresponding actions, which special case is the statement. [4], [5]

In the knowledge bases of *Wi*-concepts contact to the "etalon" images of the actions ways (in speech or impellent forms) to maintain the actions efficiency according to the existing rules and/or existing laws in mutual relations of the person with other people and an environment.

The basic knowledge of the person as syntax-logic images conjunctions of "correct" articulation are collected in neural-stacks of actions ways (skills) window in frontal areas of an impellent zone of the right and left hemisphere of human brain bark, which are directly connected by descending nervous ways to the speech

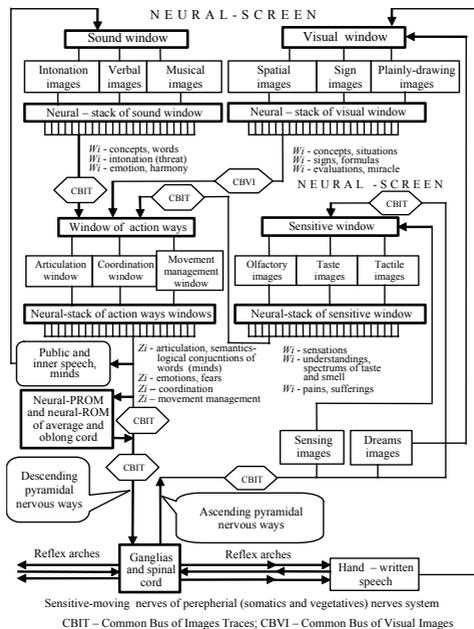


Fig. 1. The general structure of effectors networks connections of neural-stacks for various windows of the neural-screen of human brain bark.

mechanism (a mouth, a throat, tongue and lips), and also with average, oblong and a spinal cord, and through them with bodies and muscles of all body of the person. Many "sharp" direct reactions of articulation or action ways on to certain images and concepts are not realized by the person or because of absence of subconscious emotional (power) support, or are blocked by conscious and overconscious strong-willed efforts of moral-ethical behaviour skills.

The skill to think "in itself" in the form of internal speech actually is the habitual blocking of the articulations separate phases, for example, breath and lips moving, at undistinguished for itself realization of "convincing" articulation skills, although sometimes in deep meditations a certain mutter or even silent speech nevertheless sounds.

The neural-stacks of action ways window, which is subdivided into a window of an articulation, a window of coordination and a window of movement management, as it is shown on a Fig. 1, together with neural-stacks tactile window in the central area of a brain include up to 50-60 % of all neurons of brain bark that is the confirmation of importance of the movement (ways of actions) for all representatives of the biological World. Distinctive feature and a trouble of the human is filling of the neural-stacks of action (movement) ways window with "etalon" syntax-logic articulation images conjunction instead of coordination or movement skills, that sometimes even is dangerous during enforced studying of the foreign languages on the average or old human age. There are known the cases of the full loss of the coordination in space skills after the hard work in the foreign language environment of the man with bad language background.

In general, the output signals Z_i of the allocated layers of actions ways window of neural networks are the control signals which on descending pyramidal nervous ways are transferred in peripheral nervous system for realization of the actions connected to fixed concept and corresponding visual or other image. But the intellect of the person is in fact that before to start to operate he should think, accept and to explain for surrounding human his decision.

Frequently the decision is accepted by the person only intuitively, i.e. by transfer W_i from neural networks outputs from the one window on the receptors layers inputs of other windows, and the actions ways windows with output control signals Z_i of neural-stack, which "go through itself", i.e. on descending and ascending nervous ways with blocking in a spinal cord, allow at a level of body sensations images "to evaluate" an opportunity of accepted decision realization. And only after that the correctly or incorrectly choose words conjunctions during an external or internal articulation (meditations) allow to prove formally the decision and to sound it in speech for coordination of collective actions. [6]

3 The General Scheme of Decision Making in Intellectual Neural Networks

The general circuit of conclusions for any intellectual neural network includes some stages of transformations: image recognition; concept; the estimation of situation; the choice of action way; decision making; action way realization. [7]

The conclusions scheme in emotionally-oriented intellectual neural networks includes also an additional blocking-inducing chain of emotional reactions and adjusting influences on realization of habitual or unusual actions ways, which is based on known for neural networks concept and includes: the concept; emotional reaction (fears, curiosity, cares); the choice of the natural or got actions skills; the forcible-adjusting influence on decision-making (passion, famine, cold) – control and realization of actions skills.

In the intellectual emotional-oriented human neural networks the conclusions scheme includes also the parallel chain of verbal support for the "deep" conscious analysis of a situation and a logic substantiation of the decision making, that is sharply necessary at coordination of collective actions, but not always used in the real life as it is shown on Fig. 2.

Each stages in the conclusions scheme is a signaling of W_i -concepts from outputs of last layer of neurons of neural networks of the one windows and neural-stacks on inputs neural networks of other windows with the subsequent processing of the concept and concomitant factors (other types of images) in the effectors layers and the formation of the new W_i -concept-sensations and Z_i -control-ways of the actions.

Quite often the humans use W_i -emotions without the analysis of concepts or W_i -word without formation of corresponding concepts, that practically excludes an opportunity of decision-making and results in the impulsive inadequate actions or in the infinite and pointless conversations.

Basically the conclusions scheme in an ordinary life frequently does without a verbal support, and the advancing of subconscious emotions even is appreciable in comparison with overconscious intuitive estimation of current situation and the more

so with its conscious analysis on the basis of the all logic known to the person or even on the ethics which replace logic in many cases and, in particular, because of its absence.

Verbal support essentially slows down the process of conscious decision-making, because the speech images of words and phrases are compared to many images types of corresponding concepts and their relations in neural-stacks of various windows of the neural-screen and for the first hand with speech images in a neural-stack of a sound window and speech images of a neural-stack of actions ways window for articulation, where the bases of the person knowledge as corresponding types of "ready conclusions" fragments of the speech images about "correct" or "wrong" decisions in certain situations are collected, which are perceived during all life and labour activity of the human as a result of dialogue, education, training and self-training. It is should be noticed, that first two chains of the conclusions scheme, realized by subconsciousness and overconsciousness are peculiar to all representatives of the biological world.

The special problem is the conceptual (substantial) analysis of neural networks functions for overconscious estimations of situation, based on the "glorified" intuition. In abilities of an intuitive prediction the certain role is played the factors of spatial transformations of visual images in some allocated layers-processors of human brain and using of the other types of "etalon" visual images and associated with them in neural-stacks of all windows of the neural-screen.

The recognition of the most important concepts and problems in overconsciously changing pictures and the whole scripts of situations by the random or directed choosing of the comprehensible actions ways variants provide the especially successful decisions generation, which intuitively result to completely surprising

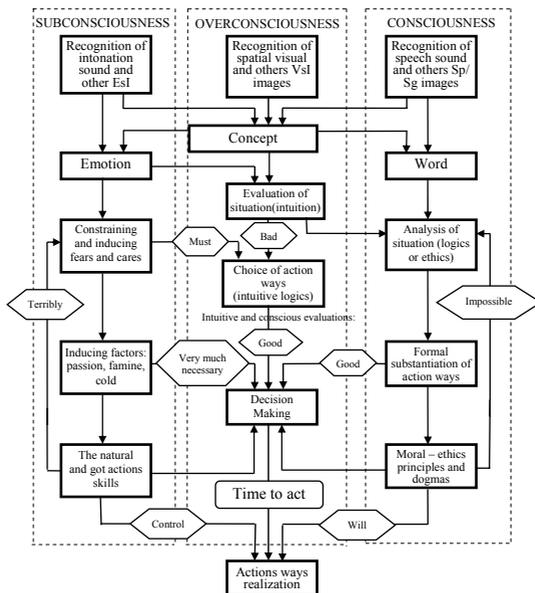


Fig. 2. The general scheme of conclusions in the emotional-oriented neural networks of human brain.

unexpressed, but to desirable results. Such "wonderful" prediction and corresponding rare, but successful concurrences of circumstances, really take place and are considered by many humans as the Miracle.

The human brain as neural networks super-parallel and highly reliable intellectual and emotionally – oriented supercomputer is the perfect creation of the Nature, but its logic organization and architecture can be realized not only on neurons, but also on some other elements or technological base in near and even the foreseeable Future.

4 The Role of Fuzzy Intuitive Estimations and Fears in Decision Making Process

The spectrum of the overconsciousness intuitive estimations-understandings of a situation by the person is insignificant. The negative intuitive estimations-understandings of current situation like "something is bad", "poorly", "very bad" and "is awful", through the mechanism of the Will and the subsystem of the Passion-Prana (Prana means the life power), as shown on United Seven-Components Model of Human Intellect (USCMHI), induce the alarm and Emotional Rise of Reserve Forces (ERRF), which provide through Neural Humoral System (NHS) inflow of forces for overcoming difficulties, problems or sufferings, and in case of powerlessness due to expected ("it is "very bad" and "is awful") cause so-called "black fear". The positive intuitive estimations-understandings of a situation, for example: "let it be", "not poorly", "there are no problems", "it is good-normally", "very well", "perfectly", also cause sometimes even the greater ERRF on the same Ring of Decision-Making (RDM): Will – Emotions (pleasure) – NHS.

The dynamical changes at the repeated independent decision of practical problems in the RDM are storing the skills of the intellect for "good" decisions acceptance on the concrete situations and thus its adaptation in the real world is provided. The conscious speech substantiation or the explanatory of the accepted decisions frequently appears difficultly formalized for the person or not explained at all, that results to misunderstanding or to mistrust on the people. The intellectual and neural-physical opportunities of the person adaptation to real environment conditions are extremely high. Undoubtedly, that the main role during adaptation plays not only the consciousness, but also overconsciousness and subconsciousness of the person and corresponding mechanisms and skills of situational management (See Fig. 2).

The development of the human intellect is provided due to the mechanism of self-training and accumulation of knowledge, skills of decision-making in contours of the mental overconsciousness (less often conscious) situational management by iterative overconsciousness (intuitive), subconscious (sensual) and conscious (formal) analysis of mental images of a subject and the actions ways, and the expected results with the subsequent formation of the overconsciousness intuitive (less often conscious logic) estimations-understandings of achievement degree level of conscious or overconscious goals.

5 The Structure of Processor-ware of Supercomputer of Human Brain

The uniform form for representation of the all traces of the objects for supervision images as traces of the structures of adjustments-associations of the neural networks, formed in neural-stacks of all windows of the neural-screen, plays the main role in the organization of coordinated work of overconsciousness, consciousnesses and subconsciousness and of their uniform interface as the Common Bus of Images Traces (CBIT). The beginning and the end of CBIT are in an impellent window of the human brain neural-screen, where as result of decision-making cycles the concrete decision on a concrete situation (See earlier Fig. 1 and Fig. 2) is realized.

CBIT is combine of the stacks of the all neural-screen windows with impellent window and its neural-stack, that is actively working in the decision-making Contour and includes: descending pyramidal ways – a spinal cord and ganglies – ascending pyramidal ways – the window and neural-stack of sensations and again an impellent window and its neural-stacks, and in some cases also visual and sound windows and their neural-stacks for the fast or direct (long) multi stages images recognition, conclusion and decision-making. The base structure of processor hardware for emotionally-oriented supercomputer of the future generations, capable to realize the basic functions of overconsciousnesses, consciousnesses and subconsciousness of the person on USCMHI, is shown on Fig. 3.

As it was shown before, the primary emotionally-oriented subsystem of overconscious-subconscious situational management can provide the continuation of human kind and the other representatives of the bio-World even in the most severe conditions of surrounding nature. Thus two base subsystems for decision-making are

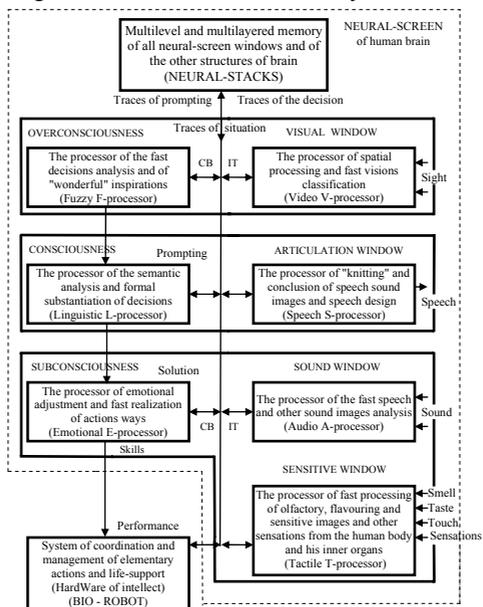


Fig. 3. The base structure of processor-ware of the emotionally and morally-oriented supercomputer of human brain.

involved: the overconsciousness, working with the induced visions of perceived images-excitations and with the packed traces of visions as adjustments-associations of the neural networks, and also the subconsciousness, operating with sound, olfactory and sensational images, that induced and packed in parameters of neural-stacks.

The important factor in conclusions of human and in the all decision-making Contour is the visions of action subject, scripts of a concrete situation and expected result. Such visions should be reflected on the neural-screen during all the time of the operative processing of the speech and other images, which are formed and transferred on CBIT. Therefore, obvious is the special role of visual support for the working of the all subsystems of human intellect – overconsciousness, consciousnesses and even subconsciousness, so the special Common Bus for visual subject and sign images (CBSI), which directly connects the processor of spatial images processing (Video V-processor) by overconsciousness with the basic processors (neural-stacks) of the brain base subsystems is obviously necessary.

The consciousness is formed only after when the person seizes speech, i.e. the specific sound signals, which fragments (words) and their various elements (statement) have the certain conditional messages, accepted in some environment – in the society and reflected in the dictionaries. Such elements of speech (language of dialogue) by the messages spectrum should differ essentially from estimations-understandings of overconsciousness.

The consciousness of the person-expert can scan the all windows of the neural-screen and to generate the mental speech image, that is marking basing on the words and statements the specific features of certain image projections of observable object, that is provided the allocation of the information and its probable further registration. That is why in some believes, that “the first was a word”. It is true, but only for the human consciousness formation.

6 Conclusions

In the paper is suggested the base biological model of thinking: “any image – action”, and also the internal articulation (idea) or the external articulation (speech) are considered as the actions ways, that naturally depend on the collective actions synchronization. Although there is the "latent" system organization of cartoids connections in receptor and effector layers of human brain, we suggest the system-technical model of the functional-oriented neural processors and of the neural coprocessors interactions organization for processing of the various type of the images and of the images traces from the knowledge bases stacks. These elements are the base for the multilevel associations of the images and their processing methods as system-psychological software, which realize all the processes of cognitive, confidential, creative cogitative and creative activities of the person (for more detail see [1]).

The advantage of the suggested structures is their flexibility and feature to create the diversified contours for the mental images processing, that provides the various decisions making depending on certain situation. The thinking and feeling robots will be highly survivable and adaptive. The specific of the suggested structures is the

complexity of the artificial intellect subsystems integration in the real practice, because it requires the significant material and human resources. However, the effective realization of the intellectual systems is impossible without the concrete concepts of the architectonics of thinking.

References

1. Shyrochin, V.P.: Architectonics of Thinking and Neural Intellect. Kiev, Junior (2004) 558 p. (in Russian)
2. Luger, G.F.: Artificial Intelligence. Structures and Strategies for Complex Problem Solving. Addison Wesley (2002) 864 p.
3. Anderson, J.R.: The Architecture of Cognition. Cambridge University Press (2003) 406 p.
4. Barr, A., Feigenbaum, E.: Handbook of Artificial Intelligence. Los Altos, CA, William Kaufman (2001) 564 p.
5. Bekhtereva, N.P.: Magic of Brain and Labyrinthes of Life. Saint-Petersburg, "Notabene" (1999) 302 p. (in Russian)
6. Jeannerod, M.: The Cognitive Neuroscience of Action. Oxford, Backwell (1997)
7. Winograd, T., Flores, F.: Understanding Computers and Cognition. Norwood, N.J., Ablex, (1996)

Initialization by Selection for Multi Library Wavelet Neural Network Training

Wajdi bellil¹, Mohamed Othmani² and Chokri Ben Amar³

¹ Faculty of sciences, University of Gafsa, City Zarroug, Gafsa, Tunisia
wajdi.bellil@ieee.org

² Department of informatics, High Institute of Technological Studies of Gafsa, Tunisia
mohamed.othmani@ieee.org

³ Department of Electrical Engineering, University of Sfax, Tunisia
Chokri.benamar@ieee.org

Abstract. This paper presents an original architecture of Wavelet Neural Network based on multi Wavelets activation function and uses a selection method to determine a set of best wavelets whose centers and dilation parameters are used as initial values for subsequent training library WNN for one dimension and two dimensions function approximation. Every input vector will be considered as unknown functional mapping and then it will be approximated by the network.

1 Introduction

Wavelet Neural Networks (WNN) were introduced by Zhang and Benveniste [1-3] in 1992 as a combination of artificial neural networks and wavelet decomposition. WNN have recently attracted great interest, because of their advantages over radial basis function networks (RBFN) as they are universal approximators but achieve faster convergence and are capable of dealing with the so-called “curse of dimensionality.” In addition, WNN are generalized RBFN. However, the generalization performance of WNN trained by least-squares approach deteriorates when outliers are present.

The task of training WNN involves estimating parameters in the network by minimizing some cost function, a measure reflecting the approximation quality performed by the network over the parameter space in the network. The least squares (LS) approach is the most popularly used in estimating the synaptic weights which provides optimal results.

Feed forward neural networks such as multilayer perceptrons (MLP) and radial basis function networks (RBFN) have been widely used as an alternative approach to functions approximation since they provide a generic black-box functional representation and have been shown to be capable of approximating any continuous function defined on a compact set in R^n with arbitrary accuracy [4]. Following the concept of locally supported basis functions such as RBFN, a class of wavelet neural networks (WNN) which originate from wavelet decomposition in signal processing has become more popular lately [5, 6, 7, 8, 9]. In addition to the salient feature of approximating

any non-linear function, WNN outperforms MLP and RBFN due to its capability in dealing with the so-called ‘‘curse of dimensionality’’ and non-stationary signals and in faster convergence speed [10]. It has also been shown that RBFN is a special case of WNN.

This paper comprises four sections. Section 2 discusses the architecture of Multi Library Wavelet Neural Networks (MLWNN) and its performance function approximation. Section 3 contributes to Beta MLWNN and to discuss the implementation and results. Finally, Section 4 gives conclusions and summary for present research work and other possibilities of future research directions.

2 Theoretical Background

2.1 Classical Wavelet Neural Network Architecture

Wavelets occur in family of functions and each is defined by dilation a_i which controls the scaling parameter and translation t_i which controls the position of a single function, named the mother wavelet $\psi(x)$. Mapping functions to a time-frequency phase space, WNN can reflect the time-frequency properties of function. Given an n -element training set, the overall response of a WNN is:

$$\hat{y}(w) = w_0 + \sum_{i=1}^{N_p} w_i \Psi_i, \text{ where } \Psi_i = \Psi\left(\frac{x - t_i}{a_i}\right) \quad (1)$$

where N_p is the number of wavelet nodes in the hidden layer and w_i is the synaptic weight of WNN.

This can also be considered as the decomposition of a function in a weighted sum of wavelets, where each weight w_j is proportional to the wavelet coefficient scaled and shifted by a_i and t_i . This establishes the idea for wavelet networks [11, 12].

This network can be considered composed of three layers: a layer with N_i inputs, a hidden layer with N_p wavelets and an output linear neuron receiving the weighted outputs of wavelets. Both input and output layers are fully connected to the hidden layer.

2.2 Multi Library Wavelet Neural Network (MLWNN)

A MLWNN can be regarded as a function approximator which estimates an unknown functional mapping:

$$y = f(x) + \varepsilon \quad (2)$$

where f is the regression function and the error term ε is a zero-mean random variable of disturbance. Constructing a MLWNN involves two stages: First, we should

construct a wavelet library $W=\{W_1, W_2, \dots, W_n\}$ of discretely dilated and translated versions of some mothers wavelets function $\Psi^1, \Psi^2, \dots, \Psi^n$:

$$W_j = \left\{ \begin{array}{l} \Psi_i^j : \Psi_i^j(x) = \alpha_i \Psi^j(a_i(x - t_i)), \\ \alpha_i = \left(\sum_{k=1}^N [\Psi^j(a_i(x_k - t_i))]^p \right)^{-\frac{1}{2}} \\ i = 1, \dots, L \text{ and } j = 1, \dots, n \end{array} \right\} \quad (3)$$

Where x_k is the sampled input, and L is the number of wavelets in each sub library W_j . Then select the best M wavelets based on the training data from multi wavelet library W , in order to build the regression. The architecture of multi library wavelet network is given in figure 1.

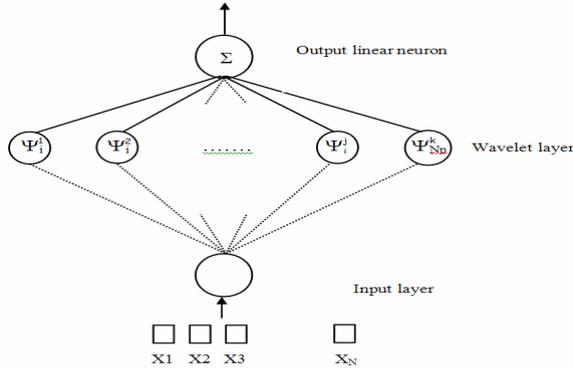


Fig. 1. MLWNN architecture.

$$\hat{y}(x) = \sum_{i \in I} w_i \Psi_i^1(x) + \sum_{i \in I} w_i \Psi_i^2(x) + \dots + \sum_{i \in I} w_i \Psi_i^n(x) \quad (4)$$

2.3 An Initialization Procedure using a Selection Method

It is very inadvisable to initialize the dilations and translations randomly, as is usually the case for the weights of a standard neural network with sigmoid activation function. In the case of wavelet neural network and due to the fact that wavelets are rapidly vanishing functions, a wavelet may be too local if its dilation parameter is too small (it may sit out of the domain of interest), if the translation parameter is not chosen appropriately.

We propose to make use of multi library wavelet using a selection method to initialize the translation and dilation parameters of wavelet networks trained using gradient-based techniques. The procedure comprises five steps:

- 1- Adapt mother wavelets support as input signal support,
- 2- Generate a multi library of wavelets, using some families of wavelets described by relation (3),
- 3- Compute the mean square error for every wavelet output,

4- Choose, from the library, the N_p wavelets that have the weakest error.

5- Use the translations and dilations of the N_p relevant wavelets as initial values and use gradient descent algorithms like least mean squares (LMS) to minimize the mean-squared error:

$$J(W) = \frac{1}{N} \sum_{i=1}^N \left(y_i - \hat{y}(W) \right)^2 \quad (5)$$

where $J(W)$ is the real output from a trained MLWNN at the fixed weight vector W .

3 BETA MLWNN for Function Approximation

In this section, we illustrate the new initialization procedure using a selection method on a multi library wavelet neural network based on Beta wavelet family and compare its effectiveness to that of the classical procedure.

3.1 Beta Wavelet Family

The Beta function [14] is defined as:

if $p > 0, q > 0, (p, q) \in \mathbb{N}$

$$\beta(x) = \begin{cases} \left(\frac{x-x_0}{x_c-x_0} \right)^p \left(\frac{x_1-x}{x_1-x_c} \right)^q & \text{if } x \in [x_0, x_1] \\ 0 & \text{else} \end{cases} \quad (6)$$

$$\text{Where, } x_c = \frac{px_1 + qx_0}{p+q}$$

We prove in [15] that all the derivatives of Beta function $\in L^2(\mathfrak{R})$, are of class C^∞ and satisfy the admissibility wavelet condition.

3.2 Example 1: 1-D Function Approximation

The first example is the approximation of a function of a single variable function, without noise, given by:

$$f(x) = \begin{cases} -2.186x - 12.864 & \text{for } x \in [-10, -2[\\ 4.246x & \text{for } x \in [-2, 0[\\ 10 \exp(-0.05x - 0.5) \sin(x(0.03x + 0.7)) & \text{for } x \in [0, 10[\end{cases} \quad (7)$$

The graph of this function is shown on Figure 2.

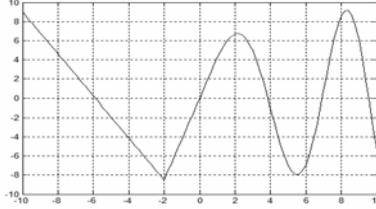


Fig. 2. The function output in the domain of interest.

First, simulations on the 1-D function approximation are conducted to validate and compare the proposed MLWNN with the classical WNN. The input x is constructed by the uniform distribution on $[-10\ 10]$. The training sequence is composed of 101 points. The performance of the model is estimated using a test set of 101 equally spaced examples different from the training set.

We define the NMSE (Normalized Mean Square Error) as evaluation criteria.

$$NMSE = \frac{1}{N} \sum_{k=1}^N \left(\hat{f}(x_k) - y_k \right)^2 \quad (8)$$

In the following, we present the results obtained with a network of 12 Beta wavelets, chosen as mother wavelets (second and third derivative of Beta function), for training network. Figure 3 shows the initial error histogram (a) obtained when the 101 input patterns are initialized with the classical architecture and the final error histogram (b) obtained when the 101 input patterns are training after 1000 iterations. Figure 4 shows the initial error histogram (c) obtained when the 101 trainings are initialized with the initialization by selection procedure using MLWNN and the final error histogram (d) obtained when the 101 input patterns are training after 1000 iterations. Comparing figures 3 and 4 shows clearly that the initialization by selection using MLWNN leads to:

- The best result in term of NMSE,
- Less scattered results both on the training set and on the test set.
- Using multi wavelet mothers as activation function gives best approximation.

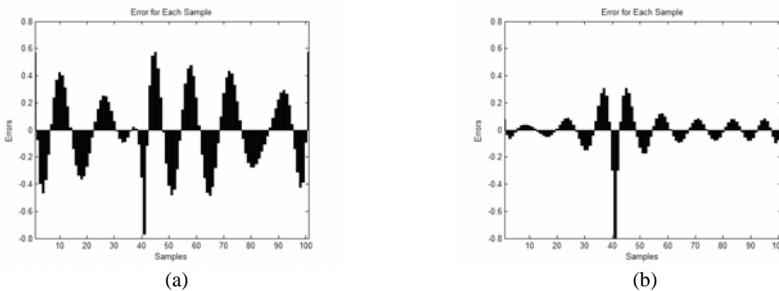


Fig. 3. (a) Initial error for each sample after initialization using classical WNN architecture.(b) Final error for each sample after initialization using classical WNN architecture after 1000 iterations.

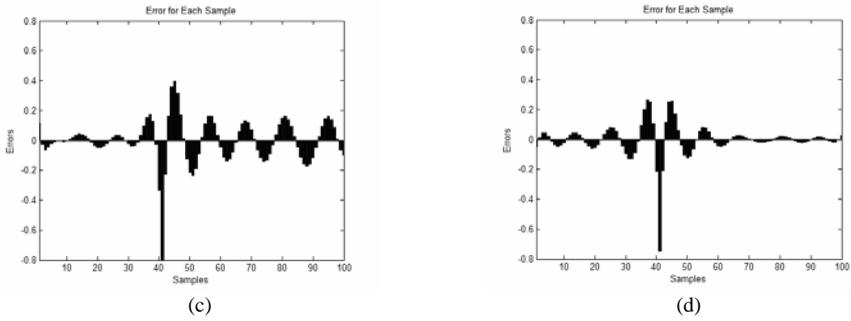


Fig. 4. (c) Initial error for each sample after initialization using MLWNN architecture. (d) Final error for each sample after initialization using MLWNN architecture after 1000 iterations.

Figure 5 shows the evolution of the NMSE according to the iteration; we can see the superiority of the proposed initialization selection algorithm based on multi wavelet library over the classical WNN based on one mother wavelet.

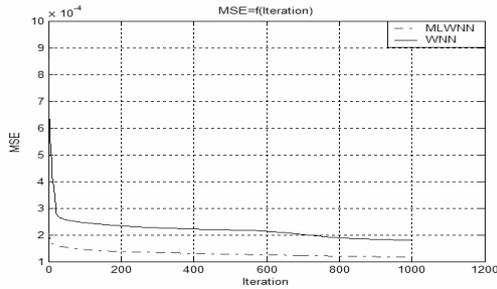


Fig. 5. Evolution of the NMSE according to the iteration.

3.3 Example 2: 2-D Function Approximation

The process to be modeled is simulated by a function of two variables without noise. The expression of this function is given by:

$$f(x_1, x_2) = \left(\frac{x_1 + 2}{2}\right)^5 \left(\frac{x_2 + 2}{2}\right)^5 \left(\frac{2 - x_1}{2}\right)^5 \left(\frac{2 - x_2}{2}\right)^5 \quad (9)$$

Figure 6 is a plot of the surface defined by relation (9).

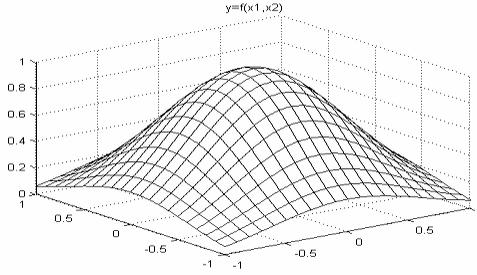


Fig. 6. 2-D data to be approximated.

In the following, we present the results obtained with a network of 9 Beta wavelets, chosen as mother wavelets (second and third derivative of Beta function), for training network. The training set contains 11×11 uniform spaced points. The test set V is constructed by 21×21 stochastic points on $[-1,1] \times [-1,1]$. Figure 7 shows the final error histogram (a) obtained when the 121 trainings are initialized with the classical architecture initialization and the final error histogram (b) obtained when the 121 trainings are initialized with a selection procedure using MLWNN.

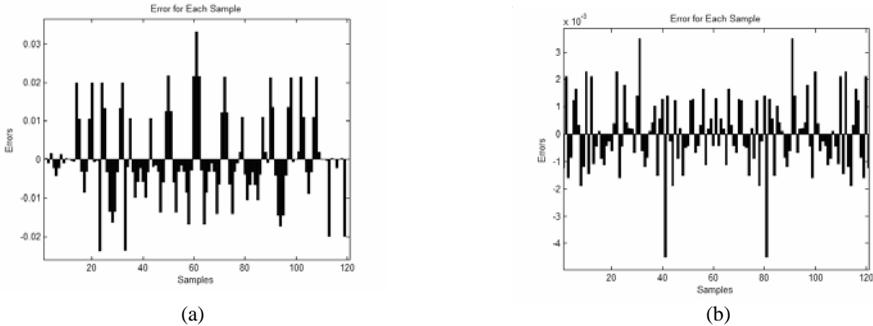


Fig. 7. (a) Final error for each sample after initialization using classical WNN architecture. (b) Final error for each sample after initialization using MLWNN architecture.

These results show that the effect of the classical WNN initialization is much smaller than when the wavelet centers and dilations are initialized by selection using a multi library WNN, used together with Beta wavelets, it makes wavelet neural network training very efficient because of the adjustable parameters of Beta function.

4 Conclusions

Wavelet networks are a class of neural networks consisting of wavelets. In this paper, we have proposed a new Initialization by Selection algorithm for Multi library Wavelet Neural Network Training for the purpose of modeling processes having a small number of inputs.

We have shown that, when used a multi library wavelet networks and a selection procedure leads to results that are much more interesting than the classical architecture initialization. The selection of “relevant” wavelets within a regular wavelet lattice can also be performed by the technique of shrinkage. However, wavelet shrinkage is usually studied with orthonormal (or biorthonormal) wavelet bases, restricted to problems of small dimension.

As future research directions, we propose to use MLWNN in the case of adaptive self tuning PID controllers. The MLWNN is needed to learn the characteristics of the plant dynamic systems and make use of it to determine the future inputs that will minimize error performance index so as to compensate the PID controller parameters.

References

1. S. Mallat, 1998. *A wavelet tour of signal processing*. academic press.
2. Q. Zang, 1997. *Wavelet Network in Nonparametric Estimation*. IEEE Trans. Neural Networks, 8(2):227-236.
3. Q. Zang et al., 1992. *Wavelet networks*. IEEE Trans. Neural Networks, vol. 3, pp. 889-898.
4. H. Boursard, Y. Kamp, 1988. *Autoassociation by multilayer perceptrons and singular values decomposition*, Biol. Cybernet. 59 (1988) 291-294.
5. A. Averbuch, D. Lazar, 1996. *Image compression using wavelet transform and multiresolution decomposition*, IEEE Trans. Image Process. 5 (1) 4-15.
6. G. Basil, J. Jiang, 1997. *Experiments on neural network implementation of LBG vector quantization*, Research Report, Depart of Computer Studies, Loughborough University.
7. G. Candotti, S. Carrato et al., 1994. *Pyramidal multiresolution source coding for progressive sequences*, IEEE Trans. Consumer Electronics 40 (4) 789-795.
8. S. Carrato, 1992. *Neural networks for image compression*, Neural Networks: Adv. and Appl. 2 ed., Gelenbe Pub, North-Holland, Amsterdam, pp. 177-198.
9. O.T.C. Chen et al., 1994. *Image compression using self-organisation networks*, IEEE Trans. Circuits Systems For Video Technol. 4 (5) (1994) 480-489.
10. M.F. Barnsley, L.P. 1993. Hurd, *Fractal Image Compression*, AK Peters Ltd.
11. Y. Oussar, 1998. *Réseaux d'ondelettes et réseaux de neurones pour la modélisation statique et dynamique de processus*, Thèse de doctorat, Université Pierre et Marie Curie.
12. C. Foucher and G. Vaucher, 2001. *Compression d'images et réseaux de neurones*, revue Valgo n°01-02, Ardèche.
13. Q. Zhang, 1997. *Using Wavelet Network in Nonparametric Estimation*, IEEE Trans. Neural Network, Vol. 8, pp.227-236.
14. C. Aouiti, M.A Alimi, and A. Maalej, 2002. *Genetic Designed Beta Basis Function Neural Networks for Multivariable Functions Approximation*, *Systems Analysis, Modeling, and Simulation*, Special Issue on Advances in Control and Computer Engineering, vol. 42, no. 7, pp. 975-1005.
15. C. Ben Amar, W. Bellil and A. Alimi. 2005-2006 *Beta Function and its Derivatives: A New Wavelet Family*. Transactions on Systems, Signals & Devices Volume 1, Number 3.
16. D. L. Donoho and I. M. Johnstone, 1995. *Adapting to unknown smoothness via wavelet shrinkage*, J. Amer. Statist. Assoc., vol. 90, no. 432, pp. 1200-1224, Dec.

ZISC Neural Network Base Indicator for Classification Complexity Estimation

Ivan Budnyk, Abdennasser Chebira and Kurosh Madani

Images, Signals and Intelligent Systems Laboratory (LISSI / EA 3956)
PARIS XII University, Senart-Fontainebleau Institute of Technology
Bat.A, Av. Pierre Point, F-77127 Lieusaint, France
budnyk@univ.kiev.ua, {chebira, madani}@univ-paris12.fr

Abstract. This paper presents a new approach for estimating task complexity using IBM© Zero Instruction Set Computer (ZISC ©). The goal is to build a neural tree structure following the paradigm “divide and rule”. The aim of this work is to define a complexity indicator-function and to hallmark its’ main features.

1 Introduction

In this paper, we present the key point of a modular neural tree structure used to solve classification problems. This modular tree structure called Tree Divide To Simplify (T-DTS) is based on the divide to conquer paradigm [1]. Complexity reduction is the key point on which the presented modular approach acts. Complexity reduction is performed not only at problem’s solution level but also at processing procedure’s level. The main idea is to reduce the complexity by splitting a complex problem into a set of simpler problems: this leads to “multi-modeling” where a set of simple models is used to sculpt a complex behavior. Thus, one of the foremost functions to be performed is the complexity estimation. The complexity estimation approach we present in this paper is based on a neurocomputer [2]. Before describing the proposed approach, we present in the second section T-DTS paradigm and then the IBM© Zero Instruction Set Computer (ZISC®) tool. In the third section we propose a new approach for complexity estimation. We validate our approach with an academic benchmark problem and study the proposed indicator function’s properties. Final section presents conclusion and further perspectives of the presented work.

2 Applied Systems

In a very large number of cases dealing with real world dilemmas and applications (system identification, industrial processes, manufacturing regulation, optimization, decision, *pattern recognition*, systems, plants safety, etc), information is available as data stored in files (databases etc...) [3]. So, efficient data processing becomes a chief condition to solve problems related to above-mentioned areas.

An issue could be model complexity reduction by splitting a complex problem into a set of simpler problems: multi-modeling where a set of simple models is used to sculpt a complex behavior ([4] and [5]). For such purpose a tree-like splitting process, based on complexity estimation, divides the problem's representative database on a set of sub-databases, constructing a specific model (dedicated processing module) for each of obtained sub-databases. That leads to a modular tree-like processing architecture including several models.

2.1 Neural Tree Modular Approach

In order to deal with real word problem, we have proposed a modular approach based on divide and conquer paradigm ([1], [3]). In this approach, Tree Divide To Simplify or T-DTS, we divide a problem in sub problems recursively and generate a neural tree computing structure. T-DTS and associated algorithm construct a tree-like evolutionary neural architecture automatically where nodes are decision units, and leafs correspond to neural based processing units ([5], [6], [7]).

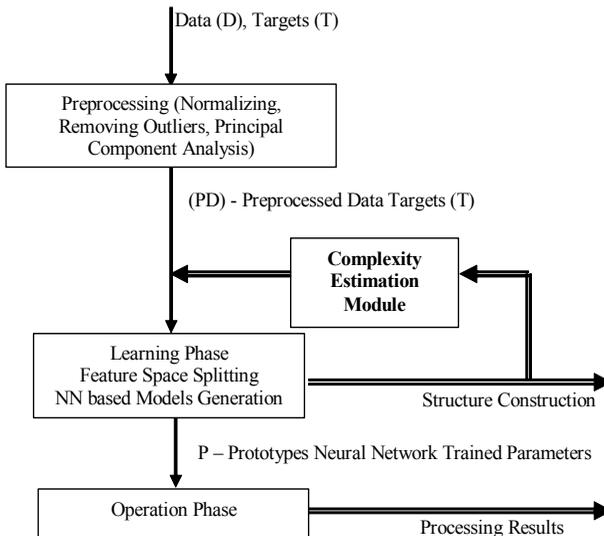


Fig. 1. General bloc diagram of T-DTS.

T-DTS includes two main operation modes. The first is the learning mode, when T-DTS system decomposes the input database and provides processing sub-structures and tools for decomposed sets of data. The second mode is the operation mode. Figure 1 gives the general bloc diagram of T-DTS operational steps. As shows this figure, T-DTS could be characterized by four main operations: “data pre-processing”, “learning process”, “generalization process” and complexity estimation module. The tree structure construction is guided mainly by the complexity estimation module. This module introduces a feedback in the learning process and control the tree computing structure. The reliability of tree model to sculpt the problem behavior is asso-

ciated mainly to the complexity estimation module. This paper focuses on this aspect and proposes a new approach based on a neurocomputer. In the following sub-section we describe ZISC® neurocomputer.

2.2 IBM(c) ZISC® Neurocomputer

ZISC® neurocomputer is a fully integrated circuit based on neural network designed for recognition and classification application which generally required supercomputing. IBM ZISC-036 ([2], [5], [8]) is a parallel neural processor based on the RCE (**Reduced Coulomb Energy** algorithm automatically adjusts the number of hidden units and converges in only few epochs. The intermediate neurons are added only when it is necessary. The influence field is then adjusted to minimize conflicting zones by a threshold) and KNN algorithms (The ***k*-nearest neighbor algorithm** - method for classifying objects based on closest training examples in the feature space. *k*-NN is a type of instance-based learning, or lazy-learning where the function is only approximated locally and all computation is deferred until classification).

Each chip is able to perform up to 250 000 recognitions per second ZISC® is the implementation of the RBF-like (Radial Basic Function) model [9]. RBF approach could be seen as mapping an N-dimensional space by prototypes. Each prototype is associated with a category and an influence field. ZISC® system implements two kinds of distance metrics called L1 and LSUP respectively. The first one (L1) corresponds to a polyhedral volume influence field and the second (LSUP) to a hyper-cubical one.

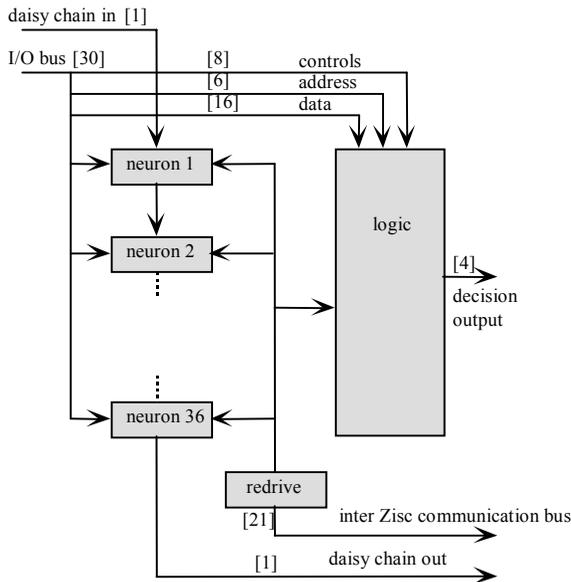


Fig. 2. IBM ZISC-036 chip's bloc diagram.

ZISC® neuron is an element, which is able:

- to memorize a prototype composed of 64 components, the associated category, an influence field and a context,
- to compute the distance, based on the selected norm (norm L1 or LSUP) between its memorized prototype and the input vector,
- to compare the computed distance with the influence fields,
- to interact with other neurons (in order to find the minimum distance, category, etc.),
- to adjust its influence field (during learning phase).

Fig 2 shows the bloc-diagram of an IBM© ZISC® chip. The next section present the complexity estimation approach based on such neurocomputer's capabilities.

3 Complexity Estimation Approach

The aim of complexity estimation is to check and measure the difficulty of a classification task, before proper processing. Classification complexity estimation is used to understand the behavior of classifiers. The most famous classification methods are based on Bayes error, the theoretical probability of classification error. However it is well known that Bayes error is difficult to compute directly. Significant part of complexity estimation methods is related to Bayes error estimation. There are two general ways to estimate Bayes error:

- indirectly [10] by proposing a measure which is a lower or higher bound of it but easier to compute than direct estimation,
- Bayes error estimation by non-parametric method and show the relation to Bayes error [11]. Other methods use space partitioning [12].

We deal with classification problems. We suppose that a database compounded of a collection of m objects associated to labels or categories is available. To estimate such database complexity we use the ZISC® as a classification tool. The goal we want to reach is not to build a classifier for this problem, but to estimate the problems' difficulty. We first used the ZISC® neurocomputer to learn this classification problem using the associated database. Then we estimate the task complexity by analyzing the generated neural network structure. We expect that a more complex problem will involve a more complex structure. The simplest neural network structure feature is the number n of neurons created during the learning phase. The following indicator is defined, where n is a parameter that reflects complexity:

$$Q = \frac{n}{m}, \quad m \geq 1, n \geq 0 \quad (1)$$

We suppose that there exists some function $n = g(.)$ that reflects problem complexity. The arguments of this function may be the signal-to-noise ratio, the dimension of the representation space, boundary non-linearity and/or database size.

In a first approach, we consider only $g(.)$ function's variations according to m axis: $g(m)$.

We suppose that our database is free of any incorrect or missing information.

On the basis on $g(m)$, a complexity indicator is defined as follow:

$$Q_i(m) = \frac{g_i(m)}{m}, m \geq 1, g_i(m) \geq 0 \quad (2)$$

We expect that for the same problem, as we enhance m , the problem seem to be less complex: more information reduces problem ambiguity. On the other hand, for problems of different and increasing complexity, Q_i indicator should have a higher value. In order to check the expected behavior of this indicator function, we have defined an academic and specific benchmark presented in the following sub-section.

3.1 Academic Benchmark Description

Basically we construct 5 databases representing a mapping of a restricted 2D space to 2 categories, (Fig. 3). Each pattern was divided into two and more equal striped sub-zones, each of them belonging to the categories 1 or 2 alternatively.

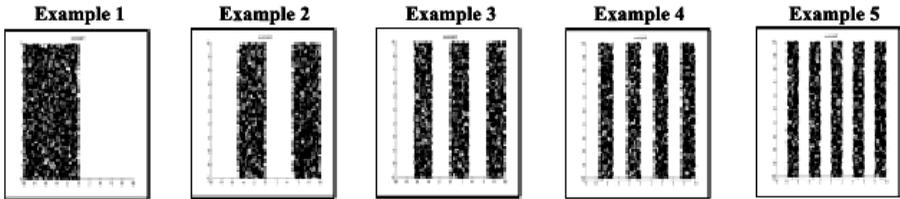


Fig. 3. Test patterns.

In *learning* mode, we create samples using randomly generated points with coordinates (x,y) . The number of samples m , in our case of uniform random distribution, naturally has an influence on the quality of the striped zones (categories) demarcation. According to the value of the first coordinate x , and according to the amount of the striped sub-zones, the appropriate category c is assigned to the sample, and such structure (x_j, y_j, c_j) sends to neurocomputer on *learning*.

The second mode is a *classification* or in other words real testing of the generalizing ZISC® neurocomputer abilities. We again, randomly and uniformly, generate m samples and their associated category. Getting classification statistics, we compute the indicator-function Q_i .

3.2 Results

The testing has been performed within:

- 2 IBM(c) ZISC® modes (LSUP/L1),
- 5 different databases with increasing complexity,
- 8 variants of m value (50, 100, 250, 500, 1000, 2500, 5000, 10000).

For each set of parameters, tests are repeated 10 times in order to get statistics and as stated to check the deviations and to get average. Totally, 800 tests have been performed.

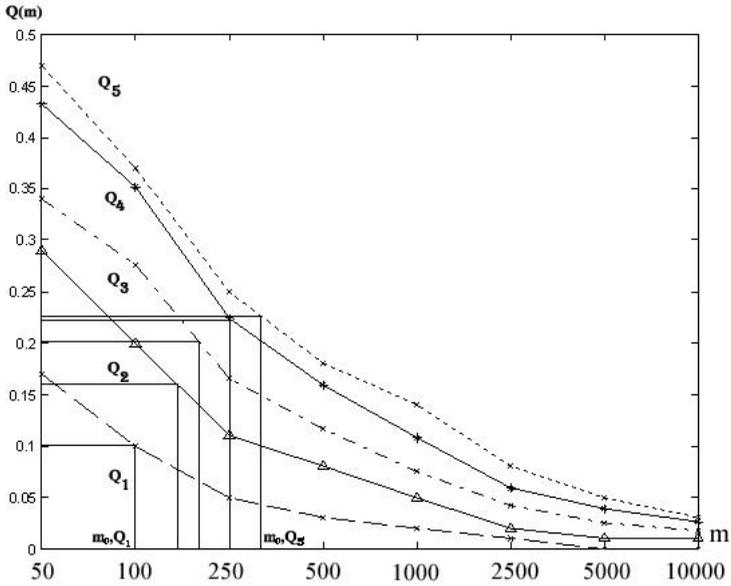


Fig. 4. LSUP ZISC's mode, $Q_i(m)$.

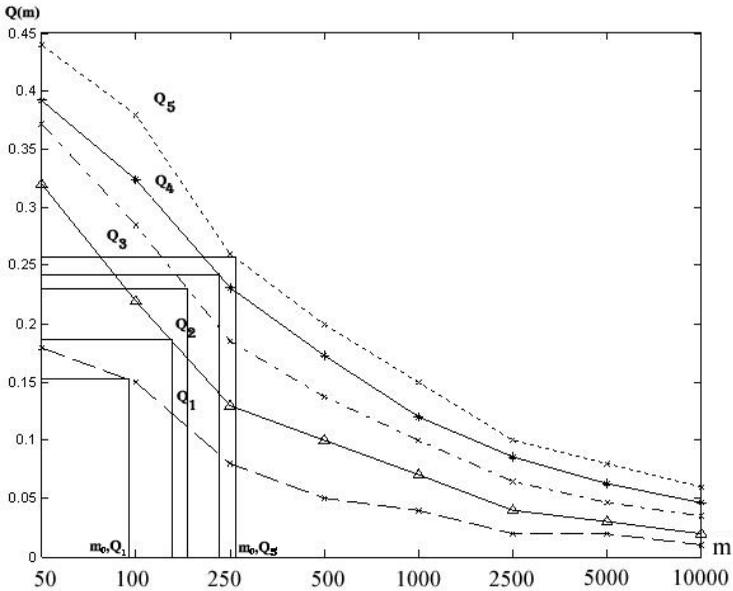


Fig. 5. L1 ZISC's mode, $Q_i(m)$.

Fig. 4 and Fig. 5 show the charts of Q_i where i is the database index or pattern index. We expect that Q_5 for 10 sub-zones reaches a higher value than Q_1 . Intuitively the problem corresponding to classification of 10 striped zones (Q_5) is more complex than for 2 (Q_1).

The chart analysis suggests that exists a point(s) m_j such as:

$$\frac{\partial^2}{(\partial m)^2} Q_i(m_j) = 0 \quad (3)$$

At such point m_j we have the following properties:

$$\forall m \geq 1, \exists \varepsilon_j > 0 : \forall m \in (m_j - \varepsilon_j; m_j) \Rightarrow \frac{\partial^2}{(\partial m)^2} Q_i(m_j) < 0, \forall m > m_j \Rightarrow \frac{\partial^2}{(\partial m)^2} Q_i(m_j) > 0 \quad (4.1)$$

or

$$\forall m \geq 1, \exists \varepsilon_j > 0 : \forall m \in (m_j - \varepsilon_j; m_j) \Rightarrow \frac{\partial^2}{(\partial m)^2} Q_i(m_j) > 0, \forall m > m_j \Rightarrow \frac{\partial^2}{(\partial m)^2} Q_i(m_j) < 0 \quad (4.2)$$

It means that there exists one or more points m_j where the second derivative of Q_i changes its sign. Then we are interesting in m_0 defined by:

$$\begin{aligned} m_0 &= \max(m_1, \dots, m_j, \dots, m_k) = m_k, \\ m_1 &< \dots < m_j < \dots < m_k \end{aligned} \quad (5)$$

Where k is the number of points m_j .

After polynomial approximation for 2 different ZISC's modes we compute the coefficients of complexity $Q_i(m_0)$. Table 1 represents the summary of the obtained results described on the Figures 4 and 5

Table 1. Coefficients of complexity for DNA sequences recognition.

ZISC's mode	LSUP		L1	
	m_0	$Q_i(m_0)$	m_0	$Q_i(m_0)$
Example 1	100	0.154	88	0.151
Example 2	170	0.182	168	0.177
Example 3	190	0.233	186	0.229
Example 4	235	0.240	229	0.239
Example 5	265	0.261	254	0.254

Main characteristic of the point m_0 is:

$$\forall m > m_0 : m \rightarrow +\infty \Rightarrow Q_i(m) \rightarrow const \quad (6)$$

In our case $const = 0$, in general not obviously $const = 0$. The feature of the second derivative sign changing is also a characteristic of success rate of the classification (Fig. 6 and Fig. 7). That supports the idea of the strong influence second derivate feature has on the complexity estimation task. That fact turn a look on the problems not

from the quantity side of complexity, but allows us to make a transitional step on the quality level. It is clearly seen that in our pattern examples, complexity of the classification is lying in the range from Example 1 (2 zones, the easiest one) till Example 5. Analysis of the plots m_{0,Q_1} (Example 1) till m_{0,Q_5} (Example 5) for related classification tasks implies the following property:

$$m_{0,Q_1} < m_{0,Q_2} < m_{0,Q_3} < m_{0,Q_4} < m_{0,Q_5} \quad (7)$$

In our particular case

$$Q_1(m_0) < Q_2(m_0) < Q_3(m_0) < Q_4(m_0) < Q_5(m_0) \quad (8)$$

In our experimental validations, the relation (6) (giving the limit of $Q_i(m)$ when m becomes $+\infty$) can be interpreted as the case where m is large comparing to m_0 meaning that the additional new data doesn't change the dynamic of the classification tasks. In other words this signifies that situation becomes more predictable regarding indicators' evolution (Fig 4 and Fig 5) and the classification rates (Fig 6 and Fig 7).

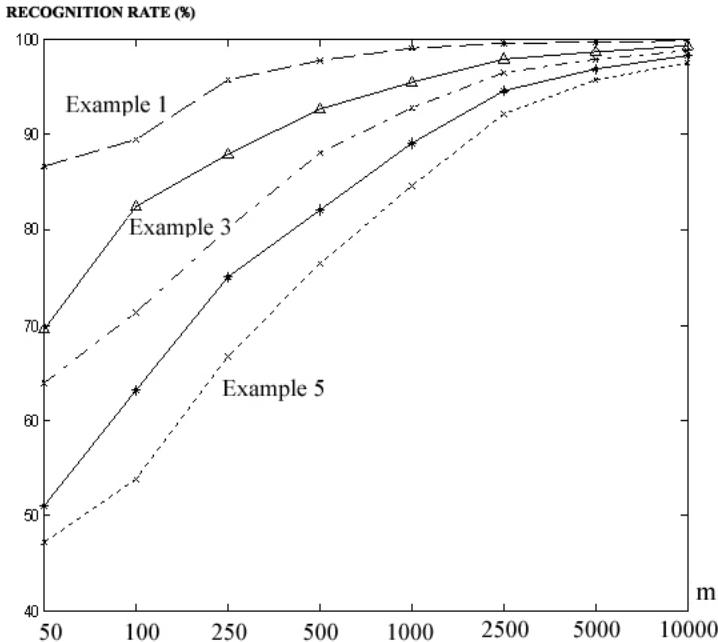


Fig. 6. Success rates of patterns' classification. Example 1 – 5. LSUP ZISC's mode.

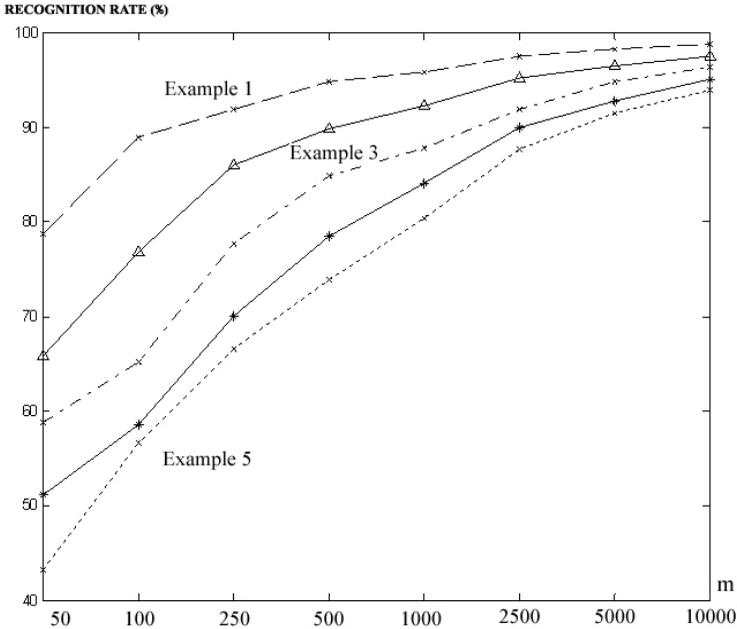


Fig. 7. Success rates of patterns' classification. Example 1 – 5. L1 ZISC's mode.

On the other hand, one can consider a particular value of m (an interesting value is m_0 for which the second derivative of $Q_i(m)$ changes the sign) making $Q_i(m_0)$ acting as a “complexity coefficient”. In our case, $Q_i(m_0)$ acts as a “checkpoint” evaluating the “stability of the classification process”. The increase of m_0 stands for the classification task's complexity increasing.

4 Conclusions

In this paper we describe a new method for complexity estimation and propose a constructed $Q(m)$ – indicator function. This approach is based on the ZISC neuro-computer. The complexity indicator is extracted from some pertinent neural network structure parameters and specifically in this paper from the number of neuron in the structure. More complex structures are related to more complex problems. The presented concept have been implemented on IBM© ZISC-036 ® massively parallel neurocomputer validated using a two-classes set of classification academic benchmarks with increasing complexity. First investigation of the second derivative sign behavior of the proposed complexity indicator allows to exhibit some interesting properties.

Perspectives of this work will be a formal description of the defined complexity indicator, the specification of other pertinent parameters and the study of their properties. We are also working on the validation of this theoretical approach to complexity

evaluation of a real-word problem in the medical area: DNA patterns classification (recognizing given a sequence of DNA the boundaries between exons and introns).

References

1. Bouyoucef, E., Chebira A., Rybnik, M., Madani, K.: 2005. Multiple Neural Network Model Generator with Complexity Estimation and self-Organization Abilities. *International Scientific Journal of Computing*, ISSN 1727-6209, vol.4, issue 3, pp.20-29.
2. Laboratory IBM France: May 15, 1998, ZISC® 036 Neurons User's Manual, Version 1.2, Component Development.
3. Madani, K., Rybnik, M., Chebira A.: 2003. Data Driven Multiple Neural Network Models Generator Based on a Tree-like Scheduler, LNCS series, Edited by: Mira, J., Prieto A., - Springer Verlag, ISBN 3-540-40210-1, pp. 382-389.
4. Jordan, M. I., Xu, L.: 1995. Convergence Results for the EM Approach to Mixture of Experts Architectures, *Neural Networks*, Vol. 8, N° 9, pp 1409-1431, Pergamon, Elsevier.
5. Madani, K., Chebira, A.: 2000. A Data Analysis Approach Based on a Neural Networks Data Sets Decomposition and it's Hardware Implementation, PKDD 2000, Lyon, France.
6. Tremiolles, G. De: March 1998. Contribution to the theoretical study of neuro-mimetic models and to their experimental validation: a panel of industrial applications, Ph.D. Report, University of PARIS XII.
7. Tremiolles, G. De, Tannhof, P., Plougouven B., Demarigny C., Madani, K.: 1997. Visual Probe Mark Inspection, using Hardware Implementation of Artificial Neural Networks, in VLSI Production, LNCS - Biological and Artificial Computation : From Neuroscience to Technology, Ed.: Mira, J., Diaz R. M., Cabestany J., Springer Verlag Berlin Heidelberg, pp. 1374-1383.
8. Laboratory IBM France: February 1995. ISC/ISA ACCELERATOR card for PC, User Manual, IBM France.
9. Park, J., Sandberg, J.W.: 1991. Universal approximation using radial basis functions network, *Neural Computation*, vol. 3. pp. 246-257.
10. Lin, J.: 1991. Divergence measures based on the Shannon entropy, *IEEE Transactions on Information Theory*, 37(1):145-151.
11. Parzen, E.: 1962. On estimation of a probability density function and mode, *Annals of Math. Statistics*, vol. 33, pp. 1065-1076.
12. Kohn, A., Nakano, L.G., Mani, V.: 1996. A class discriminability measure based on feature space partitioning, *Pattern Recognition*, 29(5):873-887.

An Improved Architecture for Cooperative and Comparative Neurons (CCNs) in Neural Network

Md. Kamrul Islam

School of Computing, Queen's University
Kingston, K7L 3N6, ON, Canada
islam@cs.queensu.ca

Abstract. The ability to store and retrieve information is critical in any type of neural network. In neural network, the memory particularly associative memory, can be defined as the one in which the input pattern leads to the response of a stored pattern (output vector) that corresponds to the input vector. During the learning phase the memory is fed with a number of input vectors that it learns and remembers and in the recall phase when some known input is presented to it, the network exactly recalls and reproduces the required output vector. In this paper, we improve and increase the storing ability of the memory model proposed in [1]. Besides, we show that there are certain instances where the algorithm in [1] does not produce the desired performance by retrieving exactly the correct vector from the memory. That is, in their algorithm, a number of output vectors can become activated from the stimulus of an input vector while the desired output is just a single correct vector. We propose a simple solution that overcomes this and can uniquely and correctly determine the output vector stored in the associative memory when an input vector is applied. Thus we provide a more general scenario of this neural network memory model consisting of memory element called Competitive Cooperative Neuron (CCN).

1 Introduction

The ability to store and retrieve information is critical in any type of neural network. In neural network, the memory, particularly associative memory, can be defined as the one in which the input pattern or vector leads to the response of a stored pattern (output vector) that corresponds to the input vector. That is, when an input vector is presented, the network recalls the corresponding output vector associated with the input vector. There are two types of associative memories: *autoassociative* and *heteroassociative* memory. In the case of autoassociative memory, both input and output vectors range over the same vector space. For example, a spelling corrector maps incorrectly spelled words (e.g. “*matual*”) to correctly spelled words (“*mutual*”). Heteroassociation involves the mapping between input and output vectors over a different vector space. For example, given a name (“*John*”) as input, the system will be able to recall its corresponding phone number (“657 – 9876”) stored in memory.

In the context of neural network, an associative memory consists of neurons (known as conventional McCulloch-Pitts [7] neurons) that are capable of processing input vector and recalling output vector. These conventional model neurons use inputs from each

source that are characterized by the amplitude of input signals. In this way each neuron can receive, process, and recall only one component of a memorized vector. Towards realizing the concept of associative memory, one of the commonly used techniques uses *correlation matrix memory* [3] which encodes all input and output vector pairs $\{y_k, x_k^T\}$ ($k = 1, 2, 3 \dots, n$) into a correlation matrix $M = \sum_{k=1}^n y_k x_k^T$. Later in the recall phase the matrix M is decoded to extract the output vector when the corresponding input vector is introduced to the network. The limitation of correlation matrix memory, in terms of memory capacity, is that it requires exactly n neurons to recall n components of a vector. In this paper we study the problem of increasing memory storage and recall capacity of a general associative memory and offer an idea that provides and ensures more storage and correct recall ability of the memory model (one layer Competitive Cooperative Neuron (CCN) network model) proposed in [1]. Our proposed method improves the architecture of the CCN network where we need only N neurons (CCNs) to store and recall NR memories where R is the number of zones (defined later) of a CCN.

The organization of the paper is as follows. First in section 2, we provide the general description of a competitive cooperative neuron (CCN), In section 3 we show how a network of such neurons can be formed and how they work by demonstrating with an example. Our main results, that is, the improvement of the CCN network model are given in section 5 which overcomes shortcomings of the model [1] (discussed in section 4). We conclude in section 6.

2 Description of a CCN

Here we provide the concise description of the CCN, the reader is referred to [1] for details. In order to increase memory storage and recall capacity of an associative memory compared to correlation matrix memory, the paper [1] introduces a novel type of model neuron called competitive and cooperative neuron (CCN) as the building block of an associative memory. This model offers two new aspects: One is that the input signals are characterized by a two-dimensional (2-D) parameter set representing the amplitude and frequency of signals. The other aspect of the CCN is that the neuron receives input signals at several distinct and autonomous receptor zones. A model of a CCN is given in Fig. 1.

The CCN consists of a number of zones R and each zone $r \in R$ collects input from many sources, $S(r) = \{S_1(r), S_2(r), S_3(r) \dots\}$. Each input signal (source) $S_i(r) = (F_i(r), A_i(r))$ has two aspects- the frequency $F_i(r) \in [0, 1]$ which encodes the information [4] and the amplitude $A_i(r) \in [0, 1]$ -the strength of the signal. Each zone is sensitive to a small range of frequencies (band). The center of the band of input r of a CCN n at time t is denoted by $B(n, r, t)$ and the tolerance level is $T(n, r, t)$. After each attempt to learn a specific memory, a band that is sufficiently close to the winning signal is preserved when the CCN fires. A zone accepts the input if the frequencies that are within its band and the amplitude of it exceeds a certain threshold $\tau(n, r, t) \in [0, 1]$. That is, a signal wins if $A_i(r) \geq \tau(n, r, t) > 0$ and $F_i(r) \in [B(n, r, t) - T(n, r, t), B(n, r, t) + T(n, r, t)]$. Each input zone propagates the winning signal to the cell body. Finally, the CCN fires if the combined amplitude of the

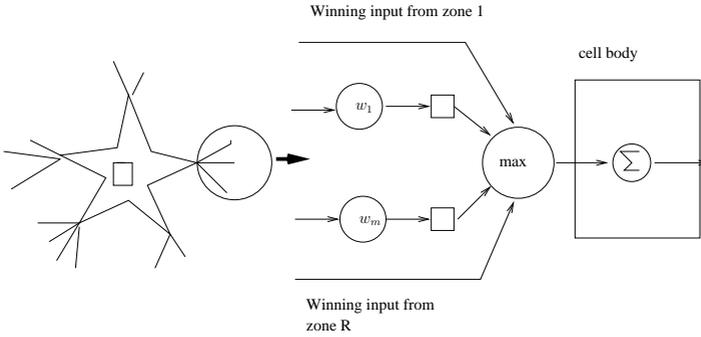


Fig. 1. A CCN Model: the CCN on the left has five autonomous zones, each of which has a narrow bandwidth of frequencies that it can detect. Each zone receives m input signals. In each zone, only the input signals that have a frequency $f_{i,r}$ that falls within the zone's bandwidth participate in the competition and the winner is the signal with the highest effective amplitude. All the winning signals are propagated to the cell's body, where they cooperate and the cell is activated if the cumulative amplitude is greater than the cell's threshold.

winning input signals from all the zones exceeds the threshold $v(n, t)$ of the CCN body, that is, $\sum_{r=1}^R A(i(r)) \geq v(n, t)$. As the CCN is activated (fired) it sets the center of the frequency band of an active zone r to its corresponding winning signal, i.e., $B(n, r, t) = F_{i(w)}(r)$. In our case, as the CCN fires we determine the output vector of it whose components are the winning signals of all the zones of the CCN. Also, the output vector can be changed by using Hebbian learning [5] protocol. Initially, we start with a CCN body threshold $v(n, t)$, which is greater than or equal to the sum of the zone thresholds, i.e., $v(n, t) \geq \sum_r \tau(n, r, t)$, so that in order for the CCN to fire, either all the zones must be active or some of them must receive a very strong signal.

When fired, the CCN decreases tolerance levels in contributing zones where the tolerance level is the maximum value of the difference between the band and the winning signal frequency that results in the firing. This is the situation when we say the CCN specializes or learns. During training, the CCN also decreases the threshold for the amplitude in active zones and also decreases the threshold of the CCN body which allows for a clean but weaker signal to activate a zone. When the CCN is trained and it receives input in some but not all zones it uses that input to recall the previous inputs to the idle zones. For example [1], if a CCN has three zones and it fired when the input was the vector that represents the triple (Red, Sweet, Strawberry), then the next time it receives only "Red" and no input from the other zones, it will fire ("Red", "Sweet", "Strawberry") provided that the amplitude of the input signal ("Red") exceeds the cell's threshold.

2.1 CCN Network Model

A simple one-layer feedforward network with three CCNs and three input sources is shown in Fig. 2. Each CCN has three receptor zones represented by the vertices of the

triangles. The number of inputs to the different zones is not necessarily the same, but it can be made the same by adding zero-weight input signals.

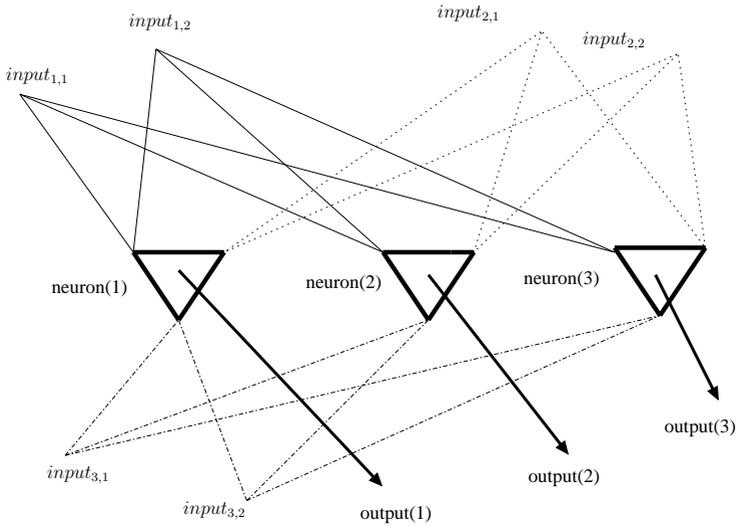


Fig. 2. A simple one-layer feedforward network with three CCNs and three input sources is shown. Each CCN has three receptor zones represented by the vertices of the triangles.

3 How a CCN Network Works

In order to understand the function of the network explained above, consider a one layer feed-forward network of CCN with N CCNs where each CCN $n \in N$ has 3 zones. Assume the centers of the frequency bands $B(n, r_1, t)$, $B(n, r_2, t)$, $B(n, r_3, t)$ of zones 1, 2 and 3 of CCN n are set to 0.150, 0.450, and 0.750 respectively. Let the tolerance level T for all zones be 0.050. The range of frequencies for each zone becomes $[B - T, B + T]$, i.e., $[0.100, 0.200]$, $[0.400, 0.500]$ and $[0.700, 0.800]$ and let the thresholds of zones 1, 2, and 3 be $\tau_1 = 0.20$, $\tau_2 = 0.15$, $\tau_3 = 0.15$ respectively and the CCN body threshold $v = 0.32$.

Assume that we want the network to store and recall vectors consisting of name, gender and id of students. Let the input vector be (“Sarah”, “Female”, “4781234”) which is represented by frequencies (0.130, 0.416, 0.725). Let 0.20 be the amplitude of each of the components of the vector. As the input vector (assuming first component of the input vector to zone 1, second component to zone 2 and so on) is applied to the network, all the zones become active and the CCN starts firing since $0.20 + 0.20 + 0.15 > 0.32$. If it does not fire then the tolerance level of inactive zones can be increased gradually to accommodate the frequency (anti Hebbian learning [5]). Now as the CCN fires the threshold to each zone is reduced to some minimum level (to some minimum

value required to activate the corresponding zone) and the threshold to the cell body is also reduced to some minimum value. Let the cell body threshold be reduced to $v = 0.18$. Now the center of the frequency band of each zone will be assigned the frequency of its winning signal, i.e., frequencies (0.130, 0.416, and 0.725) are assigned to zones 1, 2, and 3, respectively and the output vector becomes (0.130, 0.416, 0.725).

Now in the recall phase if we apply only input “Sarah” to zone 1 and no input from the other zones, the CCN will fire (because the threshold 0.20 of signal “Sarah” is greater than the CCN threshold 0.18) and produce the whole output vector (0.130, 0.416, 0.725) representing the vector (“Sarah”, “Female”, “4781234”). Therefore, if there are R (R -dimensional vector) zones in a CCN, then a single input signal to a zone will result in R recalled features ($R - 1$, if we exclude the activating input) from the other zones, which is more efficient than recalling only one feature from every input compared with the correlation matrix memory. In general, if there are N CCNs each with R zones (i.e, a CCN can store and recall a R - dimensional vector) in a one-layer feed forward CCN network then it is able to recall total NR memories. This is because, after the training is complete a signal (as given in the previous example, only input component “Sarah”) to a zone in a CCN will be strong enough to fire the CCN and recall all the other signals (components) of other zones of that CCN. As a whole, only N input signals to N CCNs will suffice to recall NR memories. On the other hand, the correlation matrix memory needs NR CCNs to recall NR memories. Therefore the achievement of performance in terms of stored-features/number-of-CCNs ratio is higher in CCN network as compared to correlation matrix memory [3].

4 Limitations in CCN

First, we show that there are certain instances where existing CCN model [1] fails to produce the expected output result. Then we offer an improvement to the architecture of the network model such that stipulated performance can be ensured. The associative network consisting of the proposed CCNs functions well if only one input vector can be attracted to at most one CCN of the network during training. This is only possible when no two CCNs have all the centers of the frequency bands ($B(n, r, t)$) are equal. The network may suffer serious limitation in manipulating (storing and recalling) data when all the centers of the frequency bands of a CCN coincide with any other CCN. Mathematically this situation can be expressed as that if there are R zones of a CCN then there exist at least two CCNs n_i and n_j such that $B(n_i, r_1, t) = B(n_j, r_1, t)$, $B(n_i, r_2, t) = B(n_j, r_2, t)$, \dots , $B(n_i, r_R, t) = B(n_j, r_R, t)$.

Under this circumstances, the network reaches a situation where the same input vector, M_j is stored in different CCNs. This is because the input, M_j stimulates and fires all those CCNs which have the same centers of frequency bands of their zones. Here we show how storage capacity decreases for the case stated above. In general, if the associative memory network has N CCNs and each CCN has R zones we can store and recall N memory vectors (total NR memories) where each memory vector M_i consists of R -components (features). In this way, we can say this is equivalent to recall exactly NR memories in total. Let S be the number of CCNs that have the same

centers of frequency bands of in their zones. This means that there is a memory vector M_s whose input can simultaneously fire S CCNs.

As M_s is stored in all the S CCNs, their centers of frequency bands will be assigned the corresponding frequencies of M_s (each component of M_s is represented by a frequency) and no other vector M_t , ($M_s \neq M_t$) can be stored in any of the S CCNs. So we have only $N - S$ CCNs left to store $N - 1$ vectors. If $S > 1$, then we can not store all the remaining vectors (remaining $N - 1$ vectors, since only one memory vector M_s is stored) to the memory. Therefore, this case does not allow us to store and recall N vectors. In the worst case, if all the N CCNs have the same centers of frequency bands then we can store only one vector in the whole network instead of N vectors. Thus the performance degrades down to $1/N$ percent which is quite worst for large values of N . In general, if S_1 is the number of CCNs having the same centers of frequency band f_1^1, \dots, f_R^1 , S_2 is the number of CCNs with the same centers of frequency band $f_1^2 \dots f_R^2$ and S_p is the number of CCNs with the same centers of frequency band f_1^p, \dots, f_R^p then we can achieve p/N percent of vectors to be stored and recalled correctly where $S_1 + S_2 + \dots + S_p = N$. The following example demonstrates such a case.

Suppose the network is required to store input vectors $\{0, 1, 0, 0\}$, $\{1, 1, 0, 0\}$, $\{1, 0, 1, 0\}$ and recall when any of the vectors is presented to the network. Assume we have a network consisting of three CCNs each with four zones. Let the first, second and third CCN's band centers are $0.1, 0.2, 0.1, 0.1$; $0.1, 0.2, 0.1, 0.1$ and $0.2, 0.2, 0.1, 0.1$, respectively. Let 0 and 1 be encoded by the frequencies 0.1 and 0.2 respectively. Therefore, we obtain the equivalent representation of the four input vectors as $\{0.1, 0.2, 0.1, 0.1\}$, $\{0.2, 0.2, 0.1, 0.1\}$ and $\{0.2, 0.1, 0.2, 0.1\}$, respectively. As we apply the input $\{0.1, 0.2, 0.1, 0.1\}$ to activate some CCN, we find all the zones of CCNs 1 and 2 become active and they (both CCNs) fire. Thus, the input vector $\{0, 1, 0, 0\}$ is stored in both of them. In this way, the two CCNs are stimulated and their centers of frequency bands are assigned the frequencies $0.1, 0.2, 0.1, 0.1$. When the second input pattern $\{0.2, 0.2, 0.1, 0.1\}$ is presented it stimulates the third CCN and causes it to fire by storing the input frequencies to the corresponding centers of frequency bands. Now for the last input there is no CCN that can be activated since all the CCNs are already attracted to the two previous input vectors. Although we have three CCNs to store and recall three vectors according to the algorithm presented in the paper [1], we cannot store more than two input patterns in the associative memory for this particular example. Thus the performance of the proposed technique degrades in this case.

5 Solution Proposed for CCNs

In this section, we outline the improvement in the architecture of the CCN neural network to remedy the situation illustrated above. This is intended so that at most one CCN can be stimulated (attracted) by a single input pattern. The modified network is shown in Fig. 3.

The main idea is to connect every CCN to all other CCNs and assign indices to them. These indices, beginning from 1 to the number of CCNs, will be assigned arbitrarily among the CCNs. It is assumed that the CCN with the lowest index has the highest

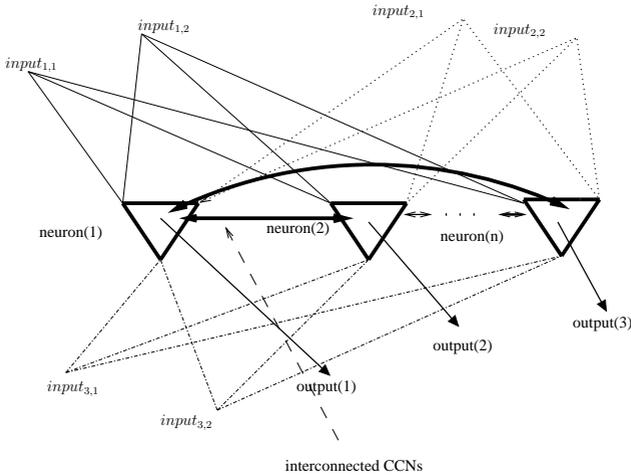


Fig. 3. An improved simple one-layer feedforward network with three CCNs and three input sources is shown. CCNs are connected to each.

priority and priority will decrease with the increase of indices. After an input pattern is applied to the network, if a CCN gets stimulated (call it *active*) then it sends its index to all other CCNs. If it is not active then it refrains from sending its index. We ensure that the highest priority active CCN will be the one to be attracted to the input if there are more than one such active CCNs.

As each active CCN sends its index to all other, every active CCN compares the index it receives from other active CCNs and if any of the indices is smaller than its own index then it does not update its center of frequency band. This means that although it is a candidate for the input to store, it withdraws its candidacy and let other higher priority CCNs to be attracted to the input. In this way, only the smallest indexed CCN wins and processes the input and changes its centers of frequency bands of its zones to the corresponding winning frequencies. In general, we define a one-to-one function $f : S \rightarrow N$, where S denotes the set of CCNs in the network. Let $S' \subseteq S$ denote the set of CCNs simultaneously attracted to an input. It is obvious that there will be exactly one $C' \in S'$ where $f(C') \neq f(C'')$ ($C'' \in S' - \{C'\}$). By following the above procedure to exchange the indices among the CCNs, we can obtain exactly one active C' which has the smallest index among the indices of the CCNs in S' since f is one-to-one. For example, in a network of 11 CCNs, if CCNs with indices 2, 7, 11 become active for some input pattern, then the CCNs 7 and 11 will withdraw because they find the index 2 is smaller. As a result, CCN 2 will take over and become stimulated and attracted to the input. The introduction of priority ensures that at any time when an input pattern is presented in the network at most one CCN will be attracted to that input. Thus we eliminate the chance of firing more than one CCNs by a single input which overcomes the problem mentioned in earlier section.

6 Conclusion

Motivated by the resemblance of a pyramidal cell [6] found in brain, the authors of [1] proposed a new type of model neuron (called CCN) to imitate the behavior of a pyramidal cell. The pyramidal cell is believed [2] to process both the frequency and the amplitude of the input signals and there is some sort of competition among inputs. Attempts are made to follow the physical structure and functional behavior of the pyramidal cell to some extent in the CCN, such as competition among the inputs and finally select the winner. In this paper, we provide an improvement to the CCN model [1] which is more generalized and can handle situation where there is a chance of getting activated more than one CCN. Furthermore, we can also increase the memory with our proposed modification to the architecture of the CCN. Thus the modified neuron model can increase the memory capacity substantially as demonstrated in this paper.

References

1. Haim Bar, W.L. Miranker, and Alexander Ambash: Competition and Cooperation in Neural processing. IEEE Transactions on Neural Networks Vol 53, No 3, March 2004
2. L.C. Rutherford, S.B. Nelson, and G.G Turrigiano; BDNF has opposite effects on the quantal amplitude of pyramidal neuron and interneuron excitatory synapses. Neuron, Vol 21, pp521-530, September, 1998.
3. <http://www.wikipedia.org/>
4. J. Singh: Great Ideas in Information Theory, Language and Cybernetics. Newyork, Dover, 1966
5. D.O Hebb: The Organization of Behavior: A Neuropsychological Theory. Newyork, Wiley, 1949
6. M. A. Arbib: The Metaphoricalbrain. Newyork: Wiley-Interscience 1972
7. S. Haykin: Neural Networks, a Comprehensive Foundation. Upper Saddle River, NJ: Prentice-Hall, 1999

Impact of Data Dimensionality Reduction on Neural Based Classification: Application to Industrial Defects

Matthieu Voiry ^{1,2}, Kurosh Madani ¹, Véronique Amarger ¹ and Joël Bernier ²

¹ Images, Signals, and Intelligent System Laboratory
(LISSI / EA 3956), Paris-XII – Val de Marne University
Senart Institute of Technology, Avenue Pierre Point, Lieusaint, 77127, France
{amarger, or madani}@univ-paris12.fr

² SAGEM REOSC
Avenue de la Tour Maury, Saint Pierre du Perray, 91280, France
{mathieu.voiry or joel.bernier}@sagem.com

Abstract. A major step for high-quality optical surfaces faults diagnosis concerns scratches and digs defects characterisation. This challenging operation is very important since it is directly linked with the produced optical component's quality. To complete optical devices diagnosis, a classification phase is mandatory since a number of correctable defects are usually present beside the potential "abiding" ones. Unfortunately relevant data extracted from raw image during defects detection phase are high dimensional. This can have harmful effect on behaviors of artificial neural networks which are suitable to perform such a challenging classification. Reducing data dimension to a smaller value can however decrease problems related to high dimensionality. In this paper we compare different techniques which permit dimensionality reduction and evaluate their possible impact on classification tasks performances.

1 Introduction

We are involved in fault diagnosis of optical devices in industrial environment. In fact, classification of detected faults is among chief phases for succeeding in such diagnosis. Aesthetic flaws, shaped during different manufacturing steps, could provoke harmful effects on optical devices' functional specificities, as well as on their optical performances by generating undesirable scatter light, which could seriously degrade the expected optical features. Taking into account the above-mentioned points, a reliable diagnosis of these defects in high-quality optical devices becomes a crucial task to ensure products' nominal specification and to enhance the production quality. Moreover, the diagnosis of these defects is strongly motivated by manufacturing process correction requirements in order to guarantee mass production (repetitive) quality with the aim of maintaining acceptable production yield.

Unfortunately, detecting and measuring such defects is still a challenging dilemma in production conditions and the few available automatic control solutions remain ineffective. That's why, in most of cases, the diagnosis is performed on the basis of a human expert based visual inspection of the whole production. However, this usual

solution suffers from several acute restrictions related to human operator's intrinsic limitations (reduced sensitivity for very small defects, detection exhaustiveness alteration due to attentiveness shrinkage, operator's tiredness and weariness due to repetitive nature of fault detection and fault diagnosis tasks).

To overcome these problems we have proposed a detection approach based on Nomarski's microscopy issued imaging [1] [2]. This method provides robust detection and reliable measurement of outward defects, making plausible a fully automatic inspection of optical products. However, the above-mentioned detection process should be completed by an automatic classification system in order to discriminate the "false" defects (correctable defects) from "abiding" (permanent) ones. In fact, because of industrial environment, a number of correctable defects (like dusts or cleaning marks) are usually present beside the potential "abiding" defects. That is why the association of a faults' classification system to the aforementioned detection module is a foremost supply to ensure a reliable diagnosis. In a precedent paper [3], we proposed a method to extract relevant data from raw Nomarski images. In the aim of effectively classify these descriptors, neural network based techniques seem appropriate because they have shown many attractive features in complex pattern recognition and classification tasks [4] [5]. But we are dealing with high dimensional data (13 and more components vectors), therefore behaviour of a number of these algorithms could be affected. To avoid this problem we are investigating different dimension reduction techniques for achieving better classification (in terms of performance and processing time).

This paper is organized as follows: in the next section, motivations for reducing data dimensionality and also SOM, CCA and CDA, three technique carrying out this task are introduced. These techniques have been tested using an experimental protocol presented in Section 3. The Section 4 deals with experiments results: first a comparison of data projection quality and an analysis of their possible impact on classification tasks are carried out. Secondly this impact is demonstrated on a classification task involving Multilayer Perceptron artificial neural network. Finally, the Section 5 will conclude this work and will give a number of perspectives.

2 Data Dimensionality Reduction Techniques

It can be found in literature, lot of examples using various dimension reduction techniques (linear or not) as a preliminary step before more refined processing, among which, Self Organizing Maps (SOM) [6;7], Curvilinear Component Analysis (CCA) [8;9] and Curvilinear Distance Analysis (CDA) [10].

2.1 The "curse of dimensionality"

Dealing with high-dimensional data indeed poses problems, known as "curse of dimensionality" [9]. First, sample number required to reach a predefined level of precision in approximation tasks, increases exponentially with dimension. Thus, intuitively, the sample number needed to properly learn problem becomes quickly much

too large to be collected by real systems, when dimension of data increases. Moreover surprising phenomena appear when working in high dimension [11] : for example, variance of distances between vectors remains fixed while its average increases with the space dimension, and Gaussian kernel local properties are also lost. These last points explain that behaviour of a number of artificial neural network algorithms could be affected while dealing with high-dimensional data. Fortunately, most real-world problem data are located in a manifold of dimension p much smaller than its raw dimension. Reducing data dimensionality to this smaller value can therefore decrease the problems related to high dimension.

2.2 Self-Organizing Maps (SOM)

Self-Organizing Map is a classical method originally proposed by Kohonen [12]. This algorithm projects multidimensional feature space into a low-dimensional presentation. Typically a SOM consists of a two dimensional grid of neurons. A vector of features is associated with each neuron. During the training phase, these vectors are tuned to represent the training data under constraint of neighbourhood conservation. Similar data are projected to the same or nearby neurons in the SOM, while different ones are mapped to neurons located further from each other, resulting in clustered data. Thus, SOM is an efficient tool for quantizing the data's space and projecting this space onto a low-dimensional space, while conserving its topology. SOM is often used in industrial engineering [13], [14] to characterize high-dimensional data or to carry out classification tasks. Unfortunately it suffers from major drawbacks: first the configuration of the topology is static and should be fixed a priori (what is efficient only for little values of projection subspace dimension), moreover the method defines only a discrete nonlinear subspace, and finally algorithm is computationally too expensive to be practically applied for projection space dimension higher than 3.

2.3 Curvilinear Component Analysis (CCA)

The goal of this technique proposed by Demartines [15] is to reproduce the topology of a n -dimension original space in a new p -dimension space (where $p < n$) without fixing any configuration of the topology. To do so, a criterion characterizing the differences between original and projected space topologies is processed:

$$E_{CCA} = \frac{1}{2} \sum_i \sum_{j \neq i} (d_{ij}^n - d_{ij}^p)^2 F(d_{ij}^p) \quad (1)$$

Where d_{ij}^n (respectively d_{ij}^p) is the Euclidean distance between vectors x_i and x_j of considered distribution in original space (resp. in projected space), and F is a decreasing function which favors local topology with respect to the global topology. This energy function is minimized by stochastic gradient descent [16]:

$$\forall i \neq j, \Delta x_i^p = \alpha(t) \frac{d_{ij}^n - d_{ij}^p}{d_{ij}^p} u(\lambda(t) - d_{ij}^p)(x_i^p - x_j^p), \quad (2)$$

Where $\alpha: \mathfrak{R}^+ \rightarrow [0;1]$ and $\lambda: \mathfrak{R}^+ \rightarrow \mathfrak{R}^+$ are two decreasing functions representing respectively a learning parameter and a neighborhood factor. CCA provides also a similar method to project, in continuous way, new points in the original space onto the projected space, using the knowledge of already projected vectors.

2.4 Curvilinear Distance Analysis (CDA)

Since CCA encounters difficulties with unfolding of very non-linear manifolds, an evolution called CDA has been proposed [17]. It involves curvilinear distances (in order to better approximate geodesic distances on the considered manifold) instead of Euclidean ones. Curvilinear distances are processed in two steps way. First is built a graph between vectors by considering k-NN, ε , or other neighbourhood, weighted by Euclidean distance between adjacent nodes. Then the curvilinear distance between two vectors is computed as the minimal distance between these vectors in the graph using Dijkstra's algorithm. Finally the original CCA algorithm is applied using processed curvilinear distances. This algorithm allows dealing with very non-linear manifolds and is much more robust against the choices of α and λ functions.

3 Experimental Validation Protocol

In order to obtain exploitable data for a classification scheme, we first needed to extract relevant information of raw Nomarski's microscopy issued images. We proposed to proceed in two steps [2]: first a detected items' images extraction phase and then an appropriated coding of the extracted images. The image associated to a given detected item is constructed considering a stripe of ten pixels around its pixels. Thus the obtained image gives an isolated (from other items) representation of the defect (e.g. depicts the defect in its immediate environment). Figure 1 gives four examples of detected items' images using the aforementioned technique. It shows different characteristic items which could be found on optical device in industrial environment.

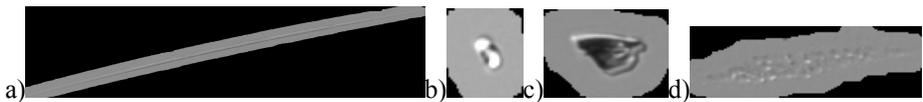


Fig. 1. Images of characteristic items: a) scratch; b) dig; c) dust; d) cleaning marks.

The information contained in such images is highly redundant. Furthermore, the generated images don't have necessarily the same dimension (typically this dimension can turn out to be thousand times as high). That is why these raw data (images) can-

not be directly processed and have to be appropriately encoded. This is done using a set of Fourier-Mellin transform issued invariants described below. The Fourier-Mellin transform of a function $f(r; \theta)$, in polar coordinates, is given by relation (1), with $q \in \mathbb{Z}$, $s = \sigma + ip \in \mathbb{C}$ (see [18]):

$$M_f(q; s) = \int_{r=0}^{\infty} \int_{\theta=0}^{2\pi} r^{s-1} \exp(-iq\theta) f(r; \theta) dr d\theta \quad (3)$$

In [19], are proposed a set of features invariant on geometric transformation:

$$I_f(q; s) = M_f(q; s) [M_f(0; \sigma)]_{\sigma}^{-s} [M_f(1; \sigma)]^{-q} |M_f(1; \sigma)|^q \quad (4)$$

In order to validate the above-presented concepts and to provide an industrial prototype, an automatic control system has been realized. It involves an Olympus B52 microscope combined with a Corvus stage, which allows scanning an entire optical component. 50x magnification is used, that leads to microscopic 1.77mm x 1.33 mm fields and 1.28 μ m x 1.28 μ m sized pixels. These facilities were used to acquire a great number of defects images. These images were coded using Fourier-Mellin transform with $\sigma=1$ and $(q, p) \in \{(q, p)/(q=0; 0 \leq p \leq P) \cup (1 \leq q \leq Q; -P \leq p \leq P)\}$ where $P=1$ and $Q=2$ (see Equation 2). Such transform provides a set of 13 features for each item. Three experiments called A, B, C were carried out, using two optical devices. Table 1 shows the different parameters corresponding to these experiments. It's important to note that, in order to avoid false classes learning, items images depicting microscopic field boundaries or two (or more) different defects are discarded from used database. First, since database C is issued from a cleaned device, it's constituted with almost only "permanent" defect. And because database B came from the measurement of the same optical device but without cleaning phase, it's constituted with the same type of "permanent" defects but also with "correctable" ones. In the aim of studying structure of space described by database when reducing its dimension, we perform some experiments. First a reduction of dimensionality from 13 (raw dimensionality) to 2 of the database B was performed using SOM, CCA and CDA, in order to compare projection quality of these three techniques. Then the entire database C was projected into the obtained space in order to evaluate the pertinence of dimensionality reduction for discrimination between "correctable" and "abiding" defects. Finally a classification task, involving aforementioned databases and Multi-layer Perceptron artificial neural network, was carried out with and without dimensionality reduction phase with the aim to demonstrate usefulness of such pre-processing phase.

Table 1. Description of the three experiments supplying studied databases.

Database	Optical Device Identifiant	Cleaning	Number of studied microscopic fields	Correspondant studied area	Number of items in the learning database
A	1	No	1178	28 cm ²	3865
B	2	No	605	14 cm ²	1910
C	2	Yes	529	12,5 cm ²	1544

4 Experimental Results and Analysis

4.1 Quality of Projection

Dimensionality reduction has been performed using the three aforementioned techniques, SOM, ACC and CDA on database B. To compare the results of the three experiments, the 2-D projections issued from CCA and CDA were processed by a SOM, using the same shape of grid (20x8) as in the SOM experiment. An important point is that SOM is just used, in these two last cases, to perform a quantization and not for dimension reduction, since it works on a 2 dimension space. Therefore, we can directly compare dimension reduction ability of the different techniques by comparing these maps with map obtained by applying SOM's algorithm on raw data. The quality evaluation of non-linear projection of the data space onto the neurons grid space is performed by studying, for each pair of neurons, the dx distance between these two neurons in the data space, versus the dy distance between these two neurons in the grid space [20]. For each couple of neurons $(i; j)$ we draw a point $(dy(i, j); dx(i, j))$ where $dx(i, j) = \|\vec{x}_i - \vec{x}_j\|$ and $dy(i, j) = \|\vec{y}_i - \vec{y}_j\|$. \vec{x}_k (resp. \vec{y}_k) is the vector of features corresponding to the k-th neuron in the data space (resp. in the grid space). If the topology of the data space is not well respected, dx is not related to dy and we obtain a diffuse cloud of points. On the contrary, if neurons organization is correct, the drawn points are almost arranged along a straight line.

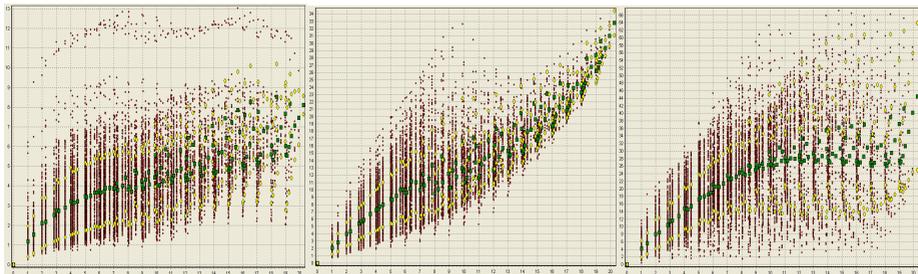


Fig. 2. dy-dx representation of the three obtained SOMs for database B (mean \square and standard deviation \diamond of dx are also represented). Left: SOM; middle: CCA; right: CDA.

First, in Figure 2, cloud of points is more diffuse for SOM than in the case of CCA, and the curve constituted by dx averages for each dy less uniformly monotonic. It reveals the fact that the CCA performs better than SOM, while approximately the same quantity is minimized. The cloud obtained for CDA is quite different because dy is related to curvilinear distance and not Euclidean one. The figure is however the same as for CCA for little dy value, because in these cases Euclidean distance is a good approximate of curvilinear one (and therefore distribution is locally linear).

4.2 Analysis of Possible Impact on Classification Tasks

We now consider the database C (only “permanent” defects) and project its items onto the three previously obtained SOMs. We perform also an equivalent experiment on raw data (13-dimension), using k-means algorithm with $k=20 \times 8=160$. Since k-means algorithm has identical behaviour as SOM, except concerning neighbourhood constraints, it has the same effect on projected items distribution but doesn’t allow visual representation. Projected items distribution after SOM (Figure 3), CCA (Figure 4) and CDA (Figure 5) dimension reduction are studied. In these figures, the equalized grey level depicts the number of projected items for each SOM’s cell (this number is also reported in the cell). In table 2 are reported some characteristic values of permanent defects distributions “homogeneity”: entropy and standard deviation of projected items number in each cell; number of empty or quasi-empty cells (<3 projected items).

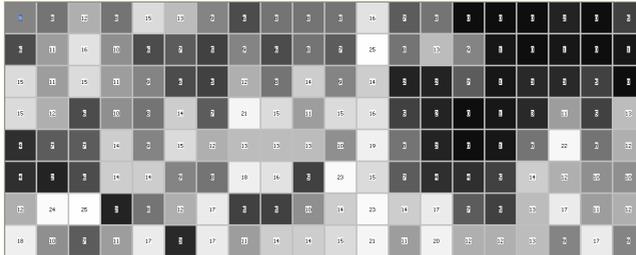


Fig. 3. Distribution of projected items in SOM map. (SOM reduction dimension).

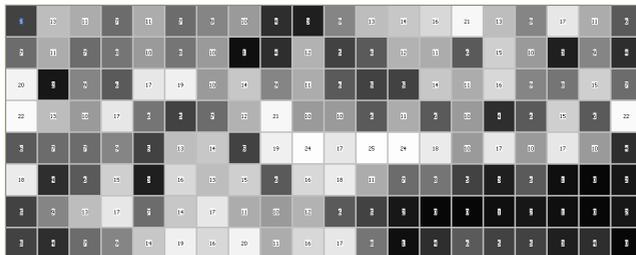


Fig. 4. Distribution of projected items in SOM map. (CCA reduction dimension).

Maps and numerical measurements for SOM and CCA are comparable and therefore these techniques are equivalent for the considered problem. CCA is however easier to

perform (no a priori knowledge or difficult choice) and provide more information (continuous projection). CDA offers the same advantages as CCA, but it seems to be more appropriate for pre-processing before classification. Corresponding map depicts indeed more specific “areas” for database C projected defects. This intuition is confirmed by numerical measurements: entropy is lower than in SOM and CCA cases (better organization), standard deviation is higher (better contrast between full and empty areas) and there are more quasi-empty cells. We think that this organization is a foremost guarantee for the dimension reduction to allow a better classification. We can also remark that results obtained with CDA are fairly similar as those with raw data; it shows that little information is lost while reducing dimensionality.

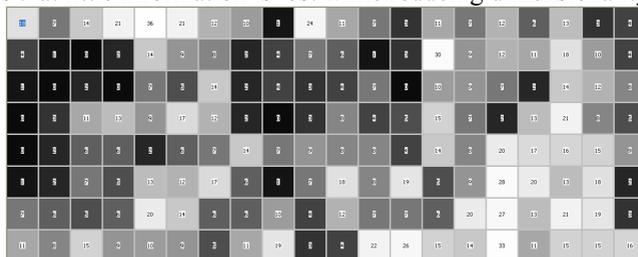


Fig. 5. Distribution of projected items in SOM map. (CDA reduction dimension).

Table 2. Different measurements characterizing the projections distribution of database C items (permanent defects).

Applied dimensionality reduction technique	Standard-deviation of projected defects distribution	Entropy of projected defects distribution	Number of empty cells	Number of cells with less than 3 defects
None	8.72	2.055	15	30
SOM	5.78	2.114	9	26
CCA	5.72	2.121	5	20
CDA	7.04	2.088	7	32

4.3 Validation on an Artificial Neural Network Based Classification

We studied a classification problem in order to evaluate pertinence of using dimension reduction before such task. First, we fixed item labels using obtained SOM with CDA dimension reduction (see Figure 5). Since database C wasn’t completely constituted of “permanent” defects (according to an expert, some dusts and cleaning marks still remain), we chose to label all SOM cells with less than 5 projected items of database C as 1: “probably correctable defects”, and the others as -1: “probably permanent defects”. Then each item from database A and B was projected into the SOM and labelled in accordance with its belonging cell, obtaining databases described in Table 3. We performed a first experiment, training a multilayer perceptron with 13 input neurons, 35 neurons in one hidden layer, and 2 output neurons (13-35-2 MLP) artificial neural network using BFGS [21] with bayesian regularization algorithm with database 2. Then a second experiment was carried out, training a 2-25-2 MLP artificial neural network with database 2 after CDA 2-dimensional space reduction.

For these two experiments, training was achieved 20 times and the generalization ability of obtained neural networks was processed using database 1. Results are presented in Table 4. Since database 1 and database 2 came from different optical devices, these generalization results are significant. These results clearly prove that the considered classification problem can be simplified, when properly reformulated in a dimension lower than its raw dimensionality and in accord with its real dimensionality.

Table 3. Description of classification databases.

Database	Coming from database	Total number of items	Label 1 items	Label -1 items
1	A	3865	1046	2816
2	B	1910	489	1421

Table 4. MLP classification performances on database 1.

CDA Reduction Dimension	Training database dimensionality	Class 1 items Well-Recognized	Class -1 items Well-Recognized	Global Performance of good classification	Global Performance Standard Deviation
No	13	71.6 %	78.0 %	76.27 %	1.37 %
Yes	2	87.4 %	96.7 %	94.16 %	0.87 %

5 Conclusion and Perspectives

A reliable diagnosis of aesthetic flaws in high-quality optical devices is a crucial task to ensure products' nominal specification and to enhance the production quality by studying the impact of the process on such defects. To ensure a reliable diagnosis, an automatic classification system is needed in order to discriminate the "correctable" defects from "abiding" ones. Unfortunately relevant data extracted from raw Nomarski image during defects detection phase are high dimensional. This can have harmful effect on behaviors of artificial neural networks which are suitable to perform such a challenging classification. Reducing the dimension of the data to a smaller value can decrease the problems related to high dimension. In this paper we have compared different techniques, SOM, CCA and CDA which permit such dimensionality reduction and evaluated their impact on classification tasks involving real industrial data. CDA seems to be the most suitable technique and we have demonstrated its ability to enhance performances in a synthetic classification task. Next phase of this work will deal with a classification task on data previously labeled by an expert.

References

1. M. Voiry, F. Houbre, V. Amarger, and K. Madani: Toward Surface Imperfections Diagnosis Using Optical Microscopy Imaging in Industrial Environment. IAR & ACD, p.139-144 (2005).

2. M. Voiry, V. Amarger, K. Madani, and F. Houbre: Combining Image Processing and Self Organizing Artificial Neural Network Based Approaches for Industrial Process Faults Clustering . 13th International Multi-Conference on Advanced Computer Systems, p.129-138 (2006).
3. M. Voiry, K. Madani, V. Amarger, and F. Houbre: Toward Automatic Defects Clustering in Industrial Production Process Combining Optical Detection and Unsupervised Artificial Neural Network Techniques. *Proceedings of the 2nd International Workshop on Artificial Neural Networks and Intelligent Information Processing - ANNIIP 2006*, p.25-34 (2006).
4. G. P. Zhang: Neural Networks for Classification: A Survey, *IEEE Trans. on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, vol. 30, no. 4, p.451-462 (2000).
5. M. Egmont-Petersen, D. de Ridder, and H. Handels : Image Processing with Neural Networks - A Review, *Pattern Recognition*, vol. 35, p.2279-2301 (2002).
6. K. Boehm, W. Broll, and M. Sokolewicz: Dynamic Gesture Recognition Using Neural Networks; A Fundament for Advanced Interaction Construction, *Proceedings of SPIE*, vol. 2177, Stereoscopic Displays and Virtual Reality Systems (1994).
7. J. Lampinen and E. Oja: Distortion Tolerant Pattern Recognition Based on Self-Organizing Feature Extraction, *IEEE Trans. on Neural Networks* vol. 6, p.539-547(1995).
8. S. Buchala, N. Davey, T. M. Gale, and R. J. Frank: Analysis of Linear and Nonlinear Dimensionality Reduction Methods for Gender Classification of Face Images, *International Journal of Systems Science* (2005).
9. M. Verleysen: Learning high-dimensional data. LFTNC'2001 - NATO Advanced Research Workshop on Limitations and Future Trends in Neural Computing (2001).
10. M. Lennon, G. Mercier, M. C. Mouchot, and L. Hubert-Moy: Curvilinear Component Analysis for Nonlinear Dimensionality Reduction of Hyperspectral Images, *Proceedings of SPIE*, vol. 4541, Image and Signal Processing for Remote Sensing VII, p.157-168 (2001).
11. P. Demartines, "Analyse de Données par Réseaux de Neurones Auto-Organisés." PhD Thesis, Institut National Polytechnique de Grenoble (1994).
12. T. Kohonen: *Self Organizing Maps*, 3rd edition ed. Berlin: Springer (2001).
13. T. Kohonen, E. Oja, O. Simula, A. Visa, and J. Kangas: Engineering Applications of the Self-Organizing Maps, *Proceedings of the IEEE*, vol. 84, no. 10, p.1358-1384 (1996).
14. J. Heikkonen and J. Lampinen: Building Industrial Applications with Neural Networks.,*Proc.European Symposium on Intelligent Techniques, ESIT'99* (1999).
15. P. Demartines and J. Héroult: Vector Quantization and Projection Neural Network, *Lecture Notes in Computer Science*, vol. 686, International Workshop on Artificial Neural Networks, p.328-333 (1993).
16. P. Demartines and J. Héroult: CCA : "Curvilinear Component Analysis", *Proceedings of 15th workshop GRETSI* (1995).
17. J. A. Lee, A. Lendasse, N. Donckers, and M. Verleysen: A Robust Nonlinear Projection Method, European Symposium on Artificial Neural Networks - ESANN'2000 (2000).
18. S. Derrode, "Représentation de Formes Planes à Niveaux de Gris par Différentes Approximations de Fourier-Mellin Analytique en vue d'Indexation de Bases d'Images." PhD Thesis , Université de Rennes I (1999).
19. F. Ghorbel: A Complete Invariant Description for Gray Level Images by the Harmonic Analysis Approach, *Pattern Recognition*, vol. 15, p.1043-1051 (1994).
20. P. Demartines and F. Blayo: Kohonen Self-Organizing Maps:Is the Normalization Necessary?, *Complex Systems*, vol. 6, no. 2, p.105-123 (1992).
21. J. E. Dennis and R. B. Schnabel: *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Englewood Cliffs, NJ: Prentice Hall (1983).

Direct and Indirect Classification of High-Frequency LNA Performance using Machine Learning Techniques

Peter C. Hung, Seán F. McLoone, Magdalena Sánchez
Ronan Farrell and Guoyan Zhang

Institute of Microelectronics and Wireless Systems, Department of Electronic Engineering
National University of Ireland Maynooth, Maynooth, Co. Kildare, Ireland
{phung, semcloone, msanchez, rfarrell, gzhang}@eeng.nuim.ie
<http://imws.eeng.nuim.ie/>

Abstract. The task of determining low noise amplifier (LNA) high-frequency performance in functional testing is as challenging as designing the circuit itself due to the difficulties associated with bringing high frequency signals off-chip. One possible strategy for circumventing these difficulties is to attempt to predict the high frequency performance measures using measurements taken at lower, more accessible, frequencies. This paper investigates the effectiveness of machine learning based classification techniques at predicting the gain of the amplifier, a key performance parameter, using such an approach. An indirect artificial neural network (ANN) and direct support vector machine (SVM) classification strategy are considered. Simulations show promising results with both methods, with SVMs outperforming ANNs for the more demanding classification scenarios.

1 Introduction

In recent years, functional testing of radio frequency integrated circuits (RFIC) has faced great challenges, especially for multi-gigahertz RF components. Two main problems exist: relaying the multi-gigahertz RF signal to the external tester without affecting the performance of tested RF circuits; building RF production testers operating in the gigahertz range that are not prohibitively expensive. While advances in technology and market requirements have seen rapid growth in high-frequency and high integration RFIC designs, testing practice has not followed suit. Indeed, reliable high-frequency testing has become the dominant factor in the cost and time-to-market of novel wireless products [1]. Consequently, developing cost-efficient testing solutions is becoming an increasingly important research topic [2-4]. Some of the proposed schemes for RFIC testing are based on an end-to-end strategy in which the output of the transmitter and the input of the receiver are linked through a loop-back connection. In this configuration, the testing of the complete system is carried out without any external stimulus by employing the on-chip digital hardware available. Unfortunately, this solution is not always applicable to all kinds of RF components. Other recent proposals for RF system testing have focused on the development of methodologies and algorithms for automated test, and Design for Testability (DfT). In Built-

In-Test (BIT), for example, additional circuitry is included that allows high frequency tests to be performed on-chip and then evaluated using lower frequency or DC external testers [3-5]. However, when considering BIT testing, issues such as area overhead for embedding and BIT power consumption can add significantly to the cost of design.

In this paper a different approach is considered. Since many RFICs show strong correlation between their responses to circuit parameter variation at different frequencies, it is hypothesised that knowledge of responses at lower frequencies may provide sufficient information to allow classification of responses at higher frequencies.

To investigate this hypothesis, testing of a low noise amplifier (LNA), a key component in modern telecommunication systems, is used as a case study. A standard 2.4 GHz design, simulated in ADS® using UMC's 0.18 μm silicon process technology, provided the data for our experiments [6]. The LNA circuit consisted of 2 bias transistors (0.18 μm channel length), 4 RF transistors (0.5 μm channel width), 4 resistors, 3 capacitors and 4 inductors and was deemed to be functioning correctly if the value of S_{21} @ 2.4 GHz was in the range 14.7 dB to 17.2 dB and faulty otherwise. S_{21} , a critical RF circuit performance measure, is essentially the gain of the amplifier.

Random circuit parameter perturbations representative of typical manufacturing process variations were generated and the value of S_{21} recorded at different frequencies. Fig. 1 shows a plot of the correlation that exists between variations in gain (S_{21} @ 2.4 GHz) and variations in the same parameter computed at other frequencies. There is a strong, but decreasing, correlation evident as the circuit excitation frequency moves away from the operating frequency.

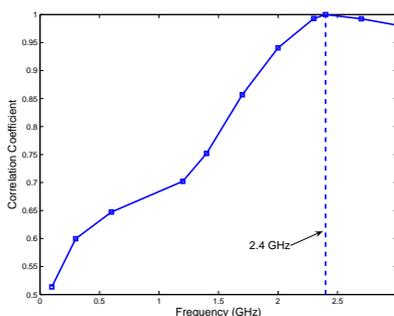


Fig. 1. Correlation coefficients between S_{21} value computed at different frequencies to the value computed at the target frequency (2.4 GHz).

Even when the correlation is high, circuit performance classification on the basis of these lower frequencies is not straightforward, as demonstrated in Fig. 2. This shows the relationship between S_{21} @ 2.4 GHz and the values computed at: (a) 2.0 GHz and (b) 0.1 GHz, respectively, and highlights the fact that even when the correlation is greater than 90% it is not possible to discriminate between the 'good' and 'bad' circuits effectively. In fact, simple thresholding on the basis of S_{21} @ 2.0 GHz leads to a misclassification rate of greater than 20%. The misclassification rate increases rapidly as the frequency is reduced and reaches 43.6% for S_{21} @ 0.1 GHz.

The rapid deterioration in performance is a result of the localised influence of some parameter variations and the complex nonlinear interaction between circuit components.

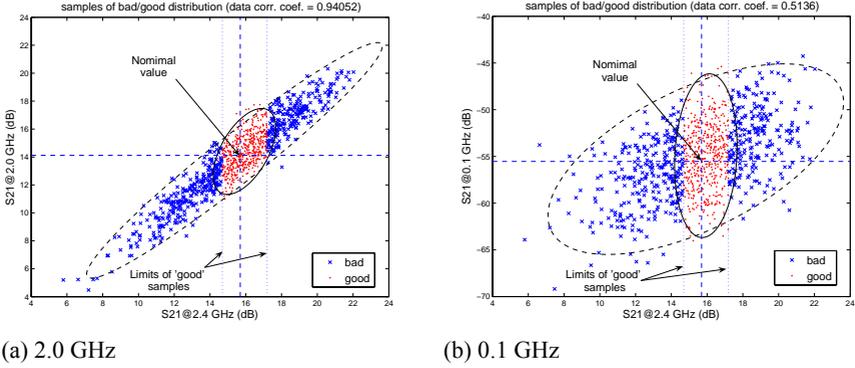


Fig. 2. S_{21} parameter relationships for the 2.4 GHz LNA model used in circuit simulations: (a) S_{21} @ 2.0 GHz and (b) S_{21} @ 0.1 GHz plotted against S_{21} at the operating frequency.

Since the shape of the frequency response of an LNA is a deterministic nonlinear function of its component parameters, better classification performance can be expected if the information from several low frequency measurements can be combined. To that end, this paper considers the possibility of classifying circuit performance using machine learning techniques. Two strategies are investigated. In the first, an Artificial Neural Network (ANN) is trained to predict the value of S_{21} @ 2.4 GHz from the values measured at other frequencies and then a thresholding rule is applied to this prediction to perform the circuit classification, while in the second, a Support Vector Machine (SVM) is trained to directly classify circuit performance on the basis of the low frequency S_{21} measurements. These two machine learning techniques and the LNA classification methodology are introduced in Section 2. The simulation study is then described in Section 3 followed by the results in Section 4. Finally, the conclusions of the study are presented in Section 5.

2 Machine Learning LNA Performance Classification

Defining the set of N low-frequency S_{21} measurements of the i^{th} LNA circuit as the feature vector \mathbf{x}_i (row vector) and the corresponding class label y_i , with $y_i = +1$ indicating ‘good’ and $y_i = -1$ indicating ‘bad’, we can generate a set of L training data examples,

$$(\mathbf{X}, \mathbf{y}) = (\mathbf{x}_1, y_1), \dots, (\mathbf{x}_i, y_i), \dots, (\mathbf{x}_L, y_L) \in \mathfrak{R}^N, \quad (1)$$

with which to train a classifier to estimate a decision function

$$f(\mathbf{x}) : \mathfrak{R}^N \rightarrow \{\pm 1\} \quad (2)$$

that can then be used to classify new LNA circuits. Here, ‘good’ and ‘bad’ are determined by a threshold function, z_{th} applied to S_{21} @ 2.4 GHz, that is

$$z_{\text{th}}(x) = \begin{cases} +1 & \text{if } 14.7 < x < 17.2 \\ -1 & \text{otherwise} \end{cases} \quad (3)$$

The decision function in (2) can be estimated directly from the training data by using, for example, a SVM classifier. Alternatively it can be estimated indirectly by first predicting the value of S_{21} @ 2.4 GHz from the feature vector,

$$g(\mathbf{x}) \rightarrow S_{21} @ 2.4\text{GHz}, \quad (4)$$

and then using a threshold function to perform the classification, that is

$$f(\mathbf{x}) : z_{\text{th}}(g(\mathbf{x})) \rightarrow \{\pm 1\}. \quad (5)$$

An ANN such as a Multilayer Perceptron can be used to learn the nonlinear mapping represented by (4).

2.1 Support Vector Machines (SVM)

SVMs, first proposed by Vladimir Vapnik in 1963 [7], are a supervised linear learning technique widely used for classification problems. They are known to perform binary classification well in many practical applications. Consider a separating hyperplane that divides two classes of data:

$$\mathbf{w} \cdot \mathbf{x} - b = 0, \quad \mathbf{w} \in \mathfrak{R}^N, b \in \mathfrak{R}, \quad (6)$$

where \mathbf{w} and b are unknown coefficients, and two additional hyperplanes that are parallel to the separating hyperplane:

$$\begin{aligned} \mathbf{w} \cdot \mathbf{x} - b &= 1 \\ \mathbf{w} \cdot \mathbf{x} - b &= -1 \end{aligned} \quad (7)$$

Defining the margin as the perpendicular distance between the parallel hyperplanes, the optimal hyperplane is the one which results in the maximum margin of separation between the two classes. Mathematically the problem can be expressed as

$$\max_{\mathbf{w}} \frac{2}{|\mathbf{w}|} \equiv \min_{\mathbf{w}} (\mathbf{w} \cdot \mathbf{w}), \quad \text{subject to } y_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1, \quad i = 1, 2, \dots, L. \quad (8)$$

This is a constrained quadratic optimisation problem whose solution \mathbf{w} has an expansion [8]

$$\mathbf{w} = \sum_i v_i \mathbf{x}_i, \quad (9)$$

where \mathbf{x}_i are the subset of the training data, referred to as support vectors, located on the parallel hyperplanes, and v_i are the corresponding weighting factors. The linear SVM (LSVM) decision function is then given by

$$f_{\text{LSVM}}(\mathbf{x}) = z_{\text{SVM}}(\mathbf{w} \cdot \mathbf{x} - b) \quad (10)$$

where z_{SVM} is defined as

$$z_{\text{SVM}}(\phi) = \begin{cases} +1 & \text{if } \phi \geq 0 \\ -1 & \text{if } \phi < 0 \end{cases} \quad (11)$$

The decision function in Eq. (10) can be rewritten as

$$f_{\text{LSVM}}(\mathbf{x}) = z_{\text{SVM}} \left(\sum_i v_i (\mathbf{x} \cdot \mathbf{x}_i) - b \right) \quad (12)$$

with the result that it is only dependent on dot products between the test data vector, \mathbf{x} , and the support vectors. This important property allows SVMs to be extended to problems where nonlinear partitions of data sets are required. This is achieved by replacing the dot products by a kernel function $k(\cdot)$ which meets the Mercer's condition [9]:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) , \quad (13)$$

thereby mapping the data into a higher dimension feature space where linear SVM classification can be performed. Note the resulting decision function, in the original data space, will be nonlinear and takes the form

$$f_{\text{SVM}}(\mathbf{x}) = z_{\text{SVM}}(r), \text{ where } r = \sum_i^L v_i \cdot k(\mathbf{x}, \mathbf{x}_i) - b \quad (14)$$

In non-separable problems where different classes of data overlap, slack variables can be introduced so that a certain amount of training error or data residing within the margin is permitted. This gives rise to a 'soft margin' optimisation function [9, 10]. To give users the ability to adjust the amount of training error allowed in the optimisation, a smoothing parameter C is incorporated into the soft margin function, with a larger C corresponding to assigning a larger penalty to errors.

The Gaussian radial basis function (RBF) defined as

$$k_{\text{RBF}}(\mathbf{x}_i, \mathbf{x}_j) = \exp \left(- \frac{|\mathbf{x}_i - \mathbf{x}_j|^2}{2\sigma^2} \right) \quad (15)$$

is a popular choice of SVM kernel and the one selected for this application. The parameter, σ , controls the width of the kernel and is determined as part of the classifier training process.

2.2 Artificial Neural Networks

Neural networks [11] are one of the best known and most commonly-used machine learning techniques. There are various configurations and structures of NNs, but all contain an array of neurons that are linked together, usually in multiple layers. In this application a single hidden layer Multilayer Perceptron (MLP) topology is chosen because of its universal function approximation capabilities, good generalisation properties and the availability of robust efficient training algorithms [12].

The output of a single hidden layer MLP can be written as a linear combination of sigmoid functions (i.e. neurons),

$$g(\mathbf{x}, \mathbf{w}_{\text{NN}}) = b^h + \sum_i w_i^h \text{sig}_i(\mathbf{x}), \quad (16)$$

where

$$\text{sig}_i(\mathbf{x}) = \frac{1}{1 + \exp(\mathbf{w}_i^u \cdot \mathbf{x} + b_i^u)}. \quad (17)$$

Here, w_i^h , \mathbf{w}_i^u , b_i^u , ($i = 1, 2, \dots, M$) and b^h are weights and biases which collectively form the network weights vector, \mathbf{w}_{NN} . Defining a Mean Squared Error (MSE) cost function over the training data

$$E(\mathbf{w}_{\text{NN}}) = \frac{1}{L} \sum_{p=1}^L (g(\mathbf{x}_p, \mathbf{w}_{\text{NN}}) - d_p)^2, \quad (18)$$

with d_p corresponding to the desired network output for the p^{th} training pattern (i.e. S_{21} @ 2.4 GHz), the optimum weights can be determined using gradient based optimisation techniques.

3 Simulation Study

To evaluate the potential for employing multiple low frequency S_{21} measurements to classify LNA S_{21} performance at 2.4 GHz and to compare the performance of the proposed machine learning classifiers, a Monte Carlo simulation study was undertaken using a 2.4 GHz LNA model implemented in ADS®. Uniform random variations were introduced into 38 of the model parameters to represent typical LNA manufacturing process variations and 10,000 circuit simulations performed. While in practice circuit parameters might be expected to vary normally around their nominal values, uniform distributions were chosen to give an even coverage of the LNA parameter space. Catastrophic failures, such as short-circuits, were not considered as these can be identified relatively easily using existing IC testing techniques.

For each circuit the S_{21} performance parameter was recorded at 0.1, 0.3, 0.6, 1.2, 1.4, 1.7 and 2.0 GHz and also at the operating frequency (2.4 GHz). This data was

then normalised to have zero mean and unit variance and divided into training and test data sets, each containing 5,000 samples.

Two different feature vectors were considered in the study, containing S_{21} measurements up to 1.4 GHz and 2.0 GHz, respectively, that is:

$$\mathbf{x}_{1.4} = [S_{21}^{0.1}, S_{21}^{0.3}, S_{21}^{0.6}, S_{21}^{1.2}, S_{21}^{1.4}] \quad (19)$$

and

$$\mathbf{x}_{2.0} = [S_{21}^{0.1}, S_{21}^{0.3}, S_{21}^{0.6}, S_{21}^{1.2}, S_{21}^{1.4}, S_{21}^{1.7}, S_{21}^{2.0}]. \quad (20)$$

Here, S_{21}^f denotes the value of S_{21} at f GHz. In each case the target MLP model output is $S_{21}^{2.4}$ while the target labels for the SVM classifier are given by $z_{\text{th}}(S_{21}^{2.4})$.

3.1 MLP Training

MLP training was performed using the hybrid BFGS training algorithm [12] with stopped minimisation used to prevent over-fitting [13]. The optimum number of neurons (M) was determined for each model by systematically evaluating different network sizes and selecting the network with the minimum MSE on the test data set. Training was repeated ten times for each network size to allow for random weight initialisations and the best set of weights recorded in each case.

The optimum network sizes and resulting model fit, measured in terms of the correlation with the true value of $S_{21}^{2.4}$, are summarised in Table 1. For comparison purposes, the correlation between $S_{21}^{2.4}$ and the measurements at 1.4 and 2.0 GHz are also given.

Table 1. Optimum MLP classifier model dimensions and resulting model fit.

Feature vector	Network dimensions	Model fit
$\mathbf{x}_{1.4}$	MLP(5,12,1)	0.9364
$\mathbf{x}_{2.0}$	MLP(7,15,1)	0.9979
$S_{21}^{1.4}$	-	0.7537
$S_{21}^{2.0}$	-	0.9408

As expected, the exploitation of multiple frequencies results in much better predictability of $S_{21}^{2.4}$ than using the measurement at a single frequency. Notably, the information provided by $S_{21}^{2.0}$ is still marginally greater than the combined information provided by all measurements up to 1.4 GHz. The classification performance of these networks, when employed in the indirect LNA classifier scheme, will be reported in Section 4.

3.2 SVM Training

SVM training was performed using the Matlab® package simpleSVM [14]. The kernel width parameter σ and smoothing parameter C were fine tuned manually and optimised on the basis of classification performance on the test data set. The final values selected were $\sigma = 0.9$ and $C = 100,000$.

Initial SVM results were quite poor despite expectations of superior performance to indirect classification using MLPs (results presented in Section 4). It was determined that this was due to the bimodal distribution of the out-of-specification circuits forming the ‘bad’ class, i.e. it consists of two segments separated by the ‘good’ class, as shown in Fig. 3.

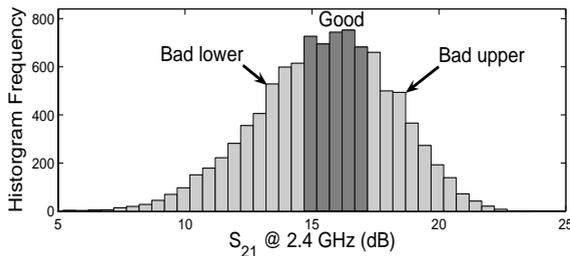


Fig. 3. Histogram of ‘good’ and ‘bad’ circuits as defined by the S_{21} performance criteria.

The *a priori* knowledge of the distribution of the ‘bad’ circuits can be taken into account by splitting the ‘bad’ samples into ‘bad lower’ and ‘bad upper’ samples, thereby introducing 3 classes – ‘bad lower’, ‘good’ and ‘bad upper’. SVM classification is then performed in two-stages. Firstly, two binary SVMs are trained, one to classify LNAs as either ‘bad lower’ or ‘not bad lower’, and one to classify LNAs as either ‘bad upper’ or ‘not bad upper’. Then the overall classification is obtained from a weighted linear combination of the individual decision functions, that is:

$$f_{\text{SVM3}} = z_{\text{SVM}}(Kr_{\text{BL}} + r_{\text{BU}}). \quad (21)$$

Here, r is as defined in Eq. (14), subscripts BL and BU represent the ‘bad lower’ and ‘bad upper’ classifiers and the constant K is a scalar which is chosen to maximise the correlation between f_{SVM3} and the true class labels over the training data.

This 3-class SVM approach is denoted SVM3 while the original two class SVM classifier will be referred to as SVM2.

4 Results

The performance of the MLP, SVM2 and SVM3 LNA classifiers was measured in terms of the following metrics computed on the test data set:

- GPR:** good pass rate - Percentage of good LNAs passed;
- BFR:** bad fail rate - Percentage of bad LNAs failed;
- FR:** failure rate - Percentage of passed LNAs incorrectly classified as ‘good’;

MCR: misclassification rate - Percentage of LNAs incorrectly classified.

Since the good pass rate (GPR) and bad fail rate (BFR) of a classifier vary as a function the classification threshold, with one increasing as the other decreases, the threshold can be adjusted to control one or other of these metrics. Here, the threshold of each classifier was adjusted to give a fixed BFR, reflecting the importance in the electronics industry of controlling the number of faulty components released to the market.

Table 2 shows the mean performance of each classifier when their thresholds were selected to give a BFR of 90% and 75%, respectively. The result for classification on the basis of single frequency measurements at 1.4 and 2.0 GHz are also included for comparison. To provide robust estimates, the metrics were computed by averaging over 100 batches of LNAs generated from the test data set using sampling with replacement. Each batch consisted of 500 ‘good’ and 500 ‘bad’ circuits randomly selected (with replacement) from a total of 1,795 ‘good’ and 3,205 ‘bad’ examples in the data set.

Table 2. Mean performance of the MLP, SVM2 and SVM3 LNA classifiers.

Inputs	Method	BFR (%) 90			75		
		GPR (%)	FR (%)	MCR (%)	GPR (%)	FR (%)	MCR (%)
$x_{1.4}$	MLP	57.11	14.88	26.45	81.03	23.55	21.99
	SVM2	44.18	18.43	27.96	77.32	24.41	11.39
	SVM3	82.30	10.82	13.85	90.85	21.56	17.08
$x_{2.0}$	MLP	99.70	9.09	5.15	100.00	19.97	12.50
	SVM2	84.40	10.57	12.80	97.33	20.41	13.83
	SVM3	97.60	9.28	6.20	99.34	20.09	12.83
$S_{21}^{1.4}$	-	19.98	33.29	45.01	47.13	34.63	38.94
$S_{21}^{2.0}$	-	55.48	15.24	27.27	84.65	22.78	20.17

Comparing the different classifiers, it can be seen that SVM3 provides the most consistent performance. Although the MLP classifier produces the best results when using frequencies up to 2.0 GHz, SVM3 is only around 2% and 0.7% behind. More importantly, when LNA classification is performed using S_{21} measurements up to 1.4 GHz only, SVM3 outperforms the MLP solution by 25% and 9% respectively. It is noted that high GPRs are always accompanied by correspondingly low values of FR and MCR. As expected, the performance of all classifiers deteriorates when only the lower frequency S_{21} measurements ($x_{1.4}$) are considered, though SVM3 and MLP still outperform the classifications obtained using a single S_{21} measurement at 2.0 GHz.

Interestingly, the SVM classifier was only able to outperform the MLP classifier when the *a priori* knowledge of the bimodal distribution of the out-of-specification LNAs was taken into account. In all cases SVM2 is substantially inferior to both the MLP and SVM3 classifiers. This suggests that while SVMs are the natural setting for classification they do not always yield the optimum results.

Although this study does not consider catastrophic IC failures, they can be identified relatively easily using other simple tests such as supply current tests. Overall, the results successfully demonstrate the usefulness of machine learning techniques for

LNA functional testing. For example, it can lower the cost of testing by extending the frequency range of existing ATE testers by as much as 70%.

5 Conclusions

Functional testing of high-frequency LNAs is becoming a prohibitively expensive and time-consuming exercise, due to the difficulties with bringing such signals off-chip. This paper proposes a novel testing strategy in which machine learning classifiers are used to predict high-frequency LNA performance by combining information from several lower frequency measurements. Promising results are obtained using both direct SVM and indirect MLP classifiers.

Acknowledgements

The authors gratefully acknowledge the financial support of Enterprise Ireland.

References

1. Ferrario, J., Wolf, R., Ding, H.: Moving from mixed signal to RF test hardware development. *IEEE Int. Test Conference* (2001) 948–956
2. Lau, W. Y.: Measurement challenges for on-wafer RF-SOC test. *27th Annual IEEE/SEMI Int. Elect. Manufact. Tech. Symp.* (2002) 353–359
3. Negreiros, M., Carro, L., Susin, A.: Low cost on-line testing of RF circuits. *10th IEEE Int. On-Line Testing Symp.* (2004) 73–78
4. Daskocil, D. C.: Advanced RF built in test. *AUTOTESTCON '92 IEEE Sys. Readiness Tech. Conf.* (1992) 213–217
5. Goff, M. E., Barratt, C. A.: DC to 40 GHz MMIC power sensor. *Gallium Arsenide IC Symp.* (1990) 105–108.
6. Allen, P. E., Holberg, D. R.: *CMOS analog circuit design*. 2nd edn. Oxford University Press, Oxford (2002)
7. Vapnik V., Lerner A.: Pattern recognition using generalised portrait method. *Automation and Remote Control* 24 (1963)
8. Hearst, M.A.: SVMs – a practical consequence of learning theory. *IEEE Intelligent Sys.* (1998) 18–21
9. Burges, C. J.: A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery* 2 (1998) 121–167
10. Cortes C., Vapnik V.: Support vector networks. *Machine Learning* 20 (1995) 273–297
11. Haykin, S.: *Neural Networks: A comprehensive foundation*. 2nd edn. Prentice Hall, New Jersey (1998)
12. McLoone S., Brown M., Irwin G., Lightbody G.: A hybrid linear/nonlinear training algorithm for feedforward neural networks. *IEEE Trans. on Neural Networks* 9 (1998) 669–684
13. Sjöberg, J., Ljung, L.: Overtraining, regularization, and searching for minimum with application to neural networks. *Int. J. Control* 62 (1995) 1391–1407
14. Vishwanathan S. V. N., Smola A. J., Murty M. N.: SimpleSVM. *Proc. 20th Int. Conf. Machine Learning* (2003)

Industrial Application Development using Case-based Reasoning

Miroslav Sveda and Ondrej Rysavy

Brno University of Technology, Faculty of Information Technology, Bozotechnova 2
CZ-612 66 Brno, Czech Republic

Abstract. Every design deserves decisions based on the application domain knowledge collected from previous similar implementations. The paper deals with stepwise development of a dedicated LAN-based industrial measurement application. The conception of this development stems from a knowledge preserving, graceful conversion of the original enterprise practice into a cooperative work supporting arrangement. The principal paradigm employed for this conversion is case-based reasoning augmented by rule-based support.

1 Introduction

Design decisions stem from the application domain knowledge that includes information about previous similar implementations. Such association with knowledge provokes to employ either an application domain expert or a dedicated knowledge-based system offering some support for the design of a novel application from scratch [1]. Another designing conception stems from the stepwise development of an original practice. In this case, the accumulated knowledge is maintained inside the budding system. Such approach can be successful when the subsequent changes are mild. On the contrary, a bigger development step requires utilizing some combination of both above mentioned conceptions [2]. This contribution presents a technique for reusing the amassed domain knowledge with an industrial measurement system whose architecture and service level were radically changed. Case-based reasoning represents its principal paradigm; besides, the overall strategy is augmented by rule-based reasoning and supporting mechanisms. Using this strategy, the knowledge-based subsystem can learn and follow the previously developed measurement processes. The original, manually controlled measurement process is carefully monitored by the subsystem in initial stages when the subsystem is actually taught by a practicing engineer. After that, the learned subsystem can continue almost autonomously, controlling not only the same but also the majority of similar measurements. The computer integrated measurement project for the test department of an electric motor manufacturer [4] offers an environment for the case study exemplifying this strategy.

2 Application

The original measurement system for induction motors of various types and sizes consisted of testing stands equipped with control switchboards, load generators, and test sets comprising both analogous and digital devices. Testing engineers proceeded measurements for obtaining unloaded characteristics, load characteristics, short circuit characteristics, torque-speed curves, temperature rise tests, and lifetime tests; furthermore, these characteristics were evaluated on a mainframe, stored on magnetic tapes, and printed out in form of certificates. The task was to design a LAN-based system substituting the mainframe functions, propping the test department agenda, and enabling to implement an intended co-operative work support; nevertheless, all measurement procedures and documents had to retain compatibility with previously decreed status, and production measurement of motors could not be interrupted or delayed.

3 Target System Architecture

The system architecture can be briefly described in the following way--see Fig. 1. A test department backbone LAN interconnects test supervisor stations TSs (industrial PCs in the field site) and, in the office position, server S and test evaluator stations TEs (workstations). Moreover, it provides through a plant backbone various communications with other departments. Each test supervisor station TS can either manage its fieldbus connecting test controllers TCs of a testing stand, or perform gateway functions between the fieldbus and the LAN. Each test controller monitors or serves attached digital and/or analogous sensors and, conceivably, actuators.

All application software modules were considered to be implemented consecutively. At the beginning, while preserving original manually mastered measurement routines, the test controller tasks sample data from auxiliary outputs of the sensors and the sensed actuators and transmit these data to the test supervisor station, which only records the course of the measurement. Data records, which are augmented by time stamps, serve both for generating standard characteristics measurement outputs reassigned to a test evaluator station, and for collecting domain-specific knowledge in form of case histories. The auxiliary outputs become the main outputs--with possible replacement of some sensors and actuators--in the next stage of the development when the test supervisor task commands the course of each measurement test. In this case, the testing engineer inspects the tests from the test supervisor console. The final stage of the field operation development includes remote supervision of test series by a test evaluator station employing case-based reasoning and possible contingent field attendant interactions through a test supervisor console.

The development of the office-position application software consists in porting the original programs from the mainframe to the server, improving the graphical presentations and interactive behavior of these programs, implementing a new database of motor characteristics, and initiating a workgroup service support. That part of the project, implemented by a larger group of programmers, exceeds the scope of the presented paper.

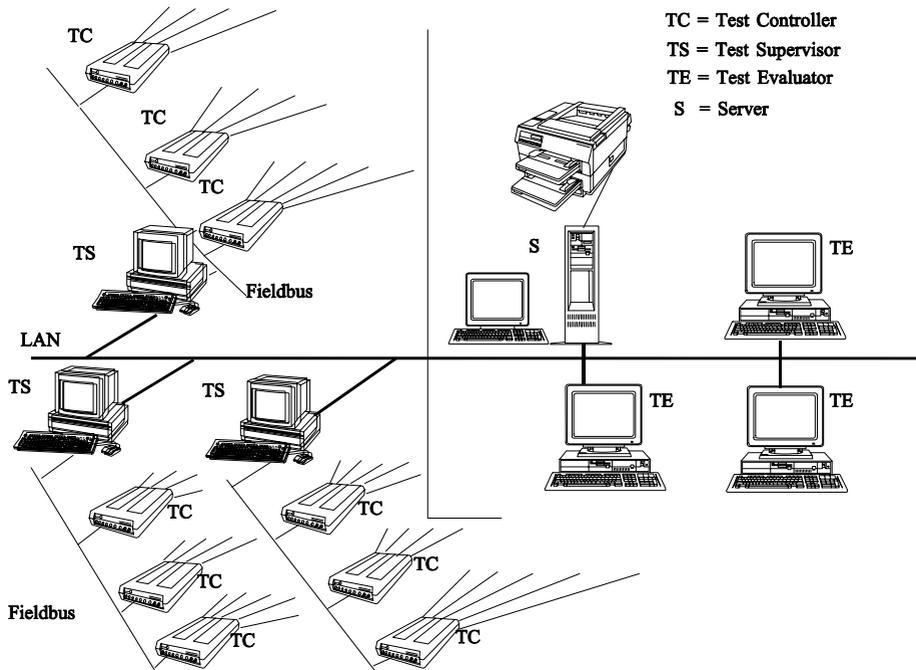


Fig. 1. Measurement system architecture.

4 Knowledge-Based Support

As mentioned earlier, the measurement process is controlled manually by experienced engineers in initial stages of the conversion. Concurrent learning process runs meanwhile collecting domain-specific knowledge about those manually controlled measurements. Knowledge-based subsystem traces expert's activities registering them with given tasks and inputs and received outputs. Thus, the learning process in initial stages facilitates a higher level control of the measurement process in later stages. So, experienced engineer can also be substituted by a novice then, without a significant worsening of the measurement quality. Case-based reasoning is the principal paradigm used for this purpose. Case-based reasoning appears to be a promising concept addressing successfully knowledge elicitation bottleneck of current knowledge-based systems [7]. Case-based reasoning differs from other rather traditional methods relying on case history. For a new problem, the case-based reasoning strives for a similar old solution. This old solution is chosen according to correspondence of the new problem to some old problem that was successfully solved by this solution. Hence, previous significant cases are gathered and saved in a case library. Case based reasoning is based on remembering the similar situation that worked in past. Elicitation means to collect those cases. Implementation represents identification of important features for the case description consisting of values of those features. A case-based reasoning system can only be as good as its case library [3]: only successful and sensibly selected old cases should be stored in the case library. The description of the

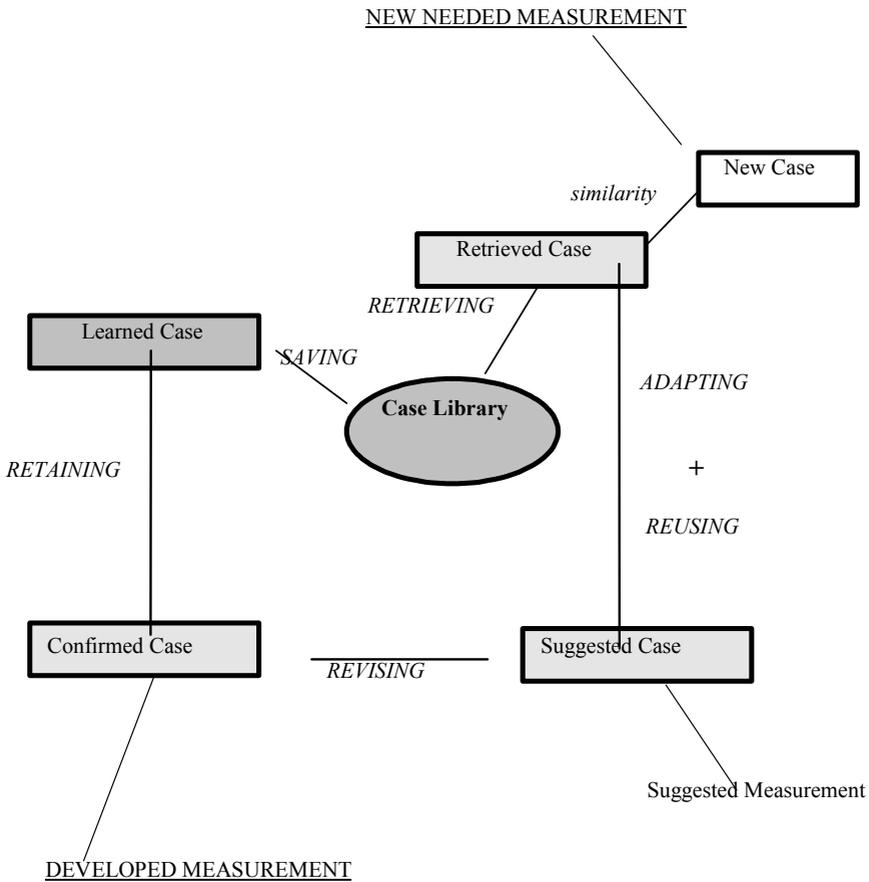


Fig. 2. Development of a new measurement procedure.

case should comprise the problem, solution of the problem, and any other information describing the context for which the solution can be reused. A feature-oriented approach is usually used for the case description.

Case library serves as a knowledge base of the case-based reasoning system. The system can learn by acquiring knowledge from the old cases. Learning is basically achieved in two ways: through accumulating new cases, and through the assignment of indexes. Solving a new case, the most similar old case is retrieved from the case library. The suggested solution of the new case is generated in conformity with this retrieved old case. Search for the similar old case from the case library represents important operation of case-based reasoning paradigm. Retrieval relies basically on two methods: nearest neighbor, and induction. The significant idea of the development of a measurement procedure is the following (Fig. 2):

- Reminding a user about earlier realized successful and verified measurement procedures in a similar context, i.e. retrieving the similar older measurement procedures from the case library;

- Facilitating reuse of the selected measurement process;
- Adaptation of the selected measurement to fit closer the new context; and
- Storing some selected measurement procedures for future reuse, after a positive evaluation process.

4.1 Case-based and Rule-based Reasoning

Case-based reasoning relies on the idea that situations are mostly repeating during the life cycle of an applied system. After some period, the most frequent situations can be identified and documented in the case library and, consequently, the case library can cover common situations. However, before the case-based paradigm can be adopted, it is necessary to control manually the measurement process for longer time, because case-based reasoning technique is not feasible at the very beginning with an empty case library.

When relying on the case-based reasoning exclusively, also the opposite problem can be encountered: after some period the case library can become huge and redundant. Registered cases represent clusters of very similar situations. Despite careful evaluation of cases before saving them in the case library, it is difficult to avoid that problem.

To compensate those insufficiencies, the case-base reasoning can be combined with rule-based support. This support has been gradually implemented. Rule-based reasoning should augment the case-based reasoning in the following situations:

- No suitable old solution can be found for a current situation in the case library and engineer hesitates about his own solution. So, rule-based module is activated. For a very restricted class of tasks, the rule-based module is capable to suggest its own solution. Once generated by this part of the framework, such a solution is then evaluated and tested more carefully. However, if the evaluation is positive, this case is later saved in the case library covering one of the gaps of the case-based module.
- Situations are similar but rarely identical. To fit closer the real situation, adaptation of the retrieved case is needed. The process of adaptation can be controlled by the rule-based paradigm, using adaptation procedures in the form of implication. Sensibly chosen meta-rules, i.e. knowledge about rules, can substantially improve the effectiveness of the system.

When relying on the case-based reasoning exclusively, also the opposite problem can be encountered: after some period the case library can become huge and redundant. Registered cases represent clusters of very similar situations. Despite careful evaluation of cases before saving them in the case library, it is difficult to avoid that problem.

4.2 Implementation Principles

For the discussed application, a case means a measurement task that consists of a sequence of measurements and its evaluation. Each measurement is described by a record containing the time stamp of the measurement and a set of attributes with their actual values sampled on inputs and outputs. There is a key parameter (either the time stamp or one item of the attributes) with prescribed values that initiate every measurement registering. The application domain knowledge consists in the definition of input/output attributes, key parameter including its measurement initiating values, output attribute values passage, and evaluation rules of input attribute value traces for a measurement task and a motor type. The evaluation usually takes a form of checking some relations among selected attributes.

5 Conclusions

Case-based reasoning, partially augmented by rule-based reasoning, has been adopted for measurement process support in the test department of an electric engine manufacturer. The chosen knowledge-based aid of the computer-based measurement system offers to preserve highly practical knowledge and experience about complex measurement procedures. Especially the case-based reasoning shows inspiring results. Role of rule-based reasoning is rather marginal. Some items of this comparison for the given application can be restated as follows:

- Case-based reasoning can work perfectly in the same or similar situation. Granularity of its knowledge base (i.e. case library) is rather coarse, with large chunks of knowledge. On the contrary, rule-based approach builds a solution gradually from a number of rather fine rules so that cases usually cover the solution entirely.
- Another advantage of the rule-based reasoning can be potentially optimal measurement procedure. On the other hand, case-based reasoning paradigm tends to rather suboptimal solutions. This paradigm relies on old and well-tried measurement procedures, updating an old solution only slightly for the purpose of adaptation.
- However, potentially rapid implementation is one of the most important advantages of the case-based reasoning. While implementation of a rule-oriented system can last years, the implementation of a case-oriented approach can be planned rather in months.
- Moreover, a needed involvement of an expert is restricted and easier for case-base reasoning paradigm. Main sources of knowledge are former measurement procedures. For this reason, substantial part of the knowledge acquisition can be automated.
- Expert cooperates on measurement description, selecting the most significant features for the description of this description and the consequent search. Expert also feels better discussing the concrete measurement procedures in the form cases, than developing general rules.

- Case library is more transparent than rule-oriented knowledge-base. Maintenance of this library is easier.

These and other factors suggest to emphasize the future development of the case-based paradigm for the similar engineering applications. Role of the rule-based support will remain restricted for the support improving the overall procedure, as mentioned above.

Acknowledgements

This paper is based on the work initially published in [5]. The knowledge-based support for that application was studied by Otakar Babka from University of Macau, Macau, Jana Freeburn from Fort Lewis College, Durango, CO, U.S.A., and the author, see [6]. The current paper restates lessons learned throughout that research while stressing more general experience utilizable for similar application domains. This work has been supported by the Czech Ministry of Education in the frame of Research Intention MSM 0021630528: Security-Oriented Research in Information Technology, and by the Grant Agency of the Czech Republic through the grants GACR 102/05/0723: A Framework for Formal Specifications and Prototyping of Information System's Network Applications and GACR 102/05/0467: Architectures of Embedded Systems Networks.

References

1. Alippi, C., Ferrero, A., Piuri, V.: Artificial Intelligence for Applications. IEEE Instrumentation and Measurement Magazine, Vol. 1, No. 2 (1998) 9-17
2. Frakes, W.B., Kang, K.: Software Reuse Research: Status and Future. IEEE Transactions on Software Engineering, Vol. 31, No.7 (2005) 529-536.
3. Kolodner, J, Case-based Reasoning. Morgan Kaufmann, San Mateo, CA, USA, (1993)
4. Machacek, J., Novotny, V., Opletal, V., Sveda, M.: Proposal of Hardware, Software, and Communication Interconnection of Automatic Measurement System for Asynchronous Motors (In Czech). Technical Report OZ 3342, VUES Brno (1991)
5. Sveda, M.: Design and Development of Industrial Measurement System--Architecture and Software. Microprocessing and Microprogramming, Vol. 40, No. 10-12 (1994) 887-890
6. Sveda, M., Babka, O., Freeburn, J.: Knowledge Preserving Development: A Case Study. IEEE Proceedings of International Conference and Workshop ECBS'97. Monterey, California, IEEE CS Press, Los Alamitos, California (1997) 347-352
7. Watson, I., Marir, F. Case-based Reasoning: A Review. The Knowledge Engineering Review, Vol. 9, No. 4. (1994)

Noisy Image Processing Using the Independent Component Analysis Algorithm AMUSE

Salua Nassabay¹, Ingo R. Keck¹, Carlos G. Puntonet¹, Juan M. Górriz²
J. Pérez de Inestroaa² and Rubén M. Clemente³

¹ Department of Architecture and Technology of Computers
Universidad Granada, ETSII, 18071 Granada, Spain

{salua,ingo,carlos}@atc.ugr.es

² Department of Signals and Communication
Universidad Granada, Granada, Spain

{javierrp,gorriz}@ugr.es

³ Department of Signals and Communication
Universidad Sevilla, Sevilla, Spain

ruben@us.es

Abstract. In this article we investigate the performance of the ICA algorithm AMUSE when applied to images contaminated by noise. The classes of noise we are using have gaussian, multiplicative and impulsive distributions. We find that AMUSE copes surprisingly well with the different types of noise, including multiplicative noise.

1 Introduction

Currently signal processing and especially the processing of images are gaining more and more importance every day. To date, different investigations have been carried out in the field of image processing whose results were compared to the Human Visual System (HVS) [1], [2], [3], to model it's capacities to adept quickly to the hugh amount of data it is constantly receiving. In order to extract the desired information from these images multistep procedures are necessary. In the first steps, the data is transformed such that its underlying structure becomes visible. The obtained data is then subject to further analysis tools in order to detect elementary components like, e.g., borders, regions, textures etc. Finally, applications are developed which aim at solving the actual problems like, e.g. recognition tasks or 3D reconstruction, etc. [4].

The present article is structured as follows: Section 2 offers a brief review of Independent Componentes Analysis (ICA) and of its most important characteristics which are exploited in Blind Source Separation (BSS). Also a brief introduction to the algorithm AMUSE (Algorithm for Multiple Unknown Signals Extraction) is given. Section 3 evaluates the performance of the algorithm when applied to data of noisy images.

1.1 Relation between ICA and Images

In the left part of figure 1 the 256 x 256 pixel image "Lena" is displayed which we have analyzed by ICA in order to obtain its typical characteristics or filters. As can be

seen in the right part of Fig. 1 these characteristics exhibit edges and other structures of interest. The characteristics were obtained by whitening the data first and by estimating afterwards the mixing-matrix \mathbf{A} by means of the fastICA algorithm. The shown patches in the right part of Fig. 1 correspond to the columns \mathbf{a}_l of the obtained mixing matrix \mathbf{A} [5].



Fig. 1. Left: original image “Lena”, 256 x 256 pixels. Right: Typical characteristics of the image, obtained applying ICA to blocks of 8 x 8 pixels.

For the processing of the image data two different approaches are usually used. The first alternative is like a local solution where the whitening-matrix $\mathbf{V}_{ZCA} = E\{\mathbf{xx}^T\}^{-1/2}$ is used to identically filter certain local regions of the data, a procedure which is similar to that occurring in the receptive fields in the retina and the lateral geniculate nucleus (LGN). As second alternative Principal Component Analysis (PCA) can be applied, so that orthogonal filters are produced that lead to uncorrelated sources. Here $\mathbf{V}_{PCA} = \mathbf{D}^{-1/2}\mathbf{E}^T$ where $\mathbf{E}\mathbf{D}\mathbf{E}^T = E\{\mathbf{xx}^T\}$ is an eigen-system of the correlation-matrix $E\{\hat{\mathbf{x}}\hat{\mathbf{x}}^T\}$. In addition PCA allows to reduce to the dimension of the problem by only selecting a subgroup of the components $\mathbf{z} = \mathbf{V}_{PCA}\mathbf{x}$, which allows us, among other things, to reduce computational costs and execution time and to lower memory consumption, etc.

Once the data has been whitened, ICA (Independent Component Analysis) is used to find the separation- or demixing-matrix \mathbf{W} such that the statistical dependence between the considered sources is minimal:

$$\hat{\mathbf{s}} = \mathbf{W}\mathbf{z} = \mathbf{W}\mathbf{V}_{PCA}\mathbf{x} = \mathbf{W}\mathbf{D}_n^{-1/2}\mathbf{D}_n^T\mathbf{x} \quad (1)$$

where \mathbf{D}_n is a diagonal matrix that contains n eigenvalues of the correlation matrix $E\{\mathbf{xx}^T\}$ and \mathbf{E}_n is the matrix having the corresponding eigenvectors in its columns.

It is important to note the similarities between the characteristics or filters found by ICA and the receptive fields of the neurons in the primary visual cortex, a similarity which eventually leads to the suggestion that the neurons are able to carry out a certain type of independent component analysis and that the receptive fields are optimized for natural images [5] [6] [7] [8].

2 Independent Component Analysis (ICA)

The concept of Independent Component Analysis was introduced by Heroult, Jutten and Ans [9] as an extension of principal component analysis. The latter is a mathematical technique that allows to project a data set to a space of characteristics whose orthogonal basis is determined such that the variance of the projections of the data onto this basis is larger than that obtained by projecting onto any other orthogonal basis. The resulting signals of a PCA transform are uncorrelated which means that the covariance or the second order cumulants, respectively, are zero.

The signals resulting from an ICA are statistically independent while no assumptions on the orthogonality of the basis vectors are made. The goal of such an ICA is then to discover a new group of meaningful signals. In order to carry out this study three hypothesis are necessary: the sources are mutually statistically independent; at most one of them has a Gaussian distribution; and the mixing model (linear, convolutive or non-linear) is known *a priori*. [9]

A lineal mixture x_1, x_2, \dots, x_n of n independent components [10], [11], is expressed mathematically by:

$$x_j = a_{j1}s_1 + a_{j2}s_2 + a_{j3}s_3 + a_{jn}s_n \text{ for all } j \quad (2)$$

where each x_j represents a mixture and each s_k represents one of the independent components. These are random variables with zero mean.

This relation can also be expressed in matrix notation: Let \mathbf{x} be the random vector having the mixtures x_1, x_2, \dots, x_n as its elements, and let \mathbf{s} be the random vector consisting of the individual sources s_1, s_2, \dots, s_n . Furthermore, consider the matrix \mathbf{A} with elements a_{ij} . Following this notation the linear mixture model can be expressed as

$$\mathbf{x} = \mathbf{A}\mathbf{s} \quad (3)$$

The ICA model is a generative model, where the observed data originates from a mixture process of the hidden original components, which are mutually independent and cannot be observed directly. This means, that only the observed data is used to recover the mixing matrix \mathbf{A} and the underlying sources \mathbf{s} .

2.1 AMUSE Algorithm

The AMUSE algorithm (Algorithm for Multiple Unknown Signals Extraction) uses temporal structures (the sources have to be uncorrelated and must have autocorrelation; no assumptions on statistical independence are necessary); it applies second order statistics with the purpose of obtaining independent components. The major motivation for the development of this algorithm was to surpass the difficulties many fourth order algorithms have when they are applied to problems with more than only one Gaussian source. [12].

The AMUSE algorithm can be formulated as follows:

1. Let $\mathbf{x}(t)$ be whitened and let $C_t^{\mathbf{x}}$ have n nondegenerated eigenvalues.

2. The eigenvalue decomposition of $C_\tau^{\mathbf{x}}$ is determined:

$$C_\tau^{\mathbf{x}} = \mathbf{W}^T D \mathbf{W} \quad (4)$$

whereas $\mathbf{W} \in O(n)$ and D is diagonal.

3. Then \mathbf{W} is the separation matrix:

$$\mathbf{W}^T = \mathbf{W}^{-1} \sim \mathbf{A} \quad (5)$$

However, the condition that all n eigenvalues are different is often strict and are a problem in real-life applications. The eigenvalues of $\text{Cov}(s_i(t), s_i(T - \tau))$ must differ significantly from each other, which is specially problematic with signals that have similar energy spectra.

3 Behavior of AMUSE when Applied to Noisy Image Data

In this section we investigate the behavior of the algorithm AMUSE when applied to the the images shown in the figure 2.

3.1 Method

As can be seen the set of images represent structures (mostly windows) which are displayed as grayscale pictures. These images consists of 256×256 pixels and each of them was previously contaminated by Gaussian, multiplicative and impulsive (also known as salt and pepper) noise. These types of noise can be seen as an own characteristic function on which the following studies have been based.



Fig. 2. Some images of structures used for the analysis.

Once having contaminated the original images with each type of noise the original and the noisy images were used to constitute the rows of the observation matrix \mathbf{X} , i.e. \mathbf{X} consisted of 64 rows and 15360 columns. For them the results have been evaluated by means of the behaviour of the filters of the different mixing matrices \mathbf{A} as well as by the typical distributions that must be preserved under the presence of noise.

Figure 3 depicts the general evaluation scheme that was used throughout this section. First, the images of the matrix \mathbf{S} (see figure 3) are used and transformed into the matrix \mathbf{X} . To these data the noise is added. From it four different observation matrices are obtained: the observation matrix of the original mixtures (\mathbf{X}_{orig}), the observation matrix contaminated with Gaussian noise \mathbf{X}_{gau} , the observation matrix contaminated with multiplicative noise (\mathbf{X}_{mul}) and the observation matrix contaminated with salt and pepper noise (\mathbf{X}_{syp}). Once the observations are created AMUSE is applied. Then the histograms are evaluated and the different results are compared with the goal to detect the filters which contain only noise.

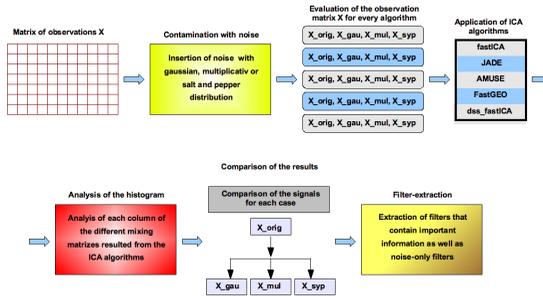


Fig. 3. General diagram. Scheme that describes the separate steps of the analysis.

3.2 Behavior of AMUSE

The results obtained from this analysis are comparatively extensive as 5 different algorithms and 3 classes of noise were used; its because of this reason that the algorithm AMUSE was used at this point of the study as it exhibits a series of particularities, especially in the context of impulsive noise. Apart from this, AMUSE was found stable no matter of the class of noise used in the data set.

First, the behavior of the bases of the mixing-matrix are investigated for each of the two cases (original and noisy signals) after dimension reduction by PCA. In this process the dimensions have been reduced to 49, 36, 25, 16, and 9 respectively. The results obtained after reducing the dimension are shown in 4 for the original signals, in 5 for the signals with Gaussian noise, in 6 for signals with multiplicative noise and in 7 for signals with salt and pepper noise.

Consider for example image 4 which presents a comparison between each of the filters while reducing the dimension, the purpose being to detect those filters which are stable and to find out if there could be a connection between the different classes of data. Independent of the class of noise, AMUSE found stable results in the filters, in where each iteration of the comparison between the different dimensional reductions also presented a concentration of stable filters in first 8 positions and mostly also in the last 3 or 4 filters. In these figures, the red frames show the stable filters (first and last) that stay throughout each reduction of dimension, arriving to obtain finally a reduction

to 9 dimensions in that the filters appear clean from noise and clearly describe important information.

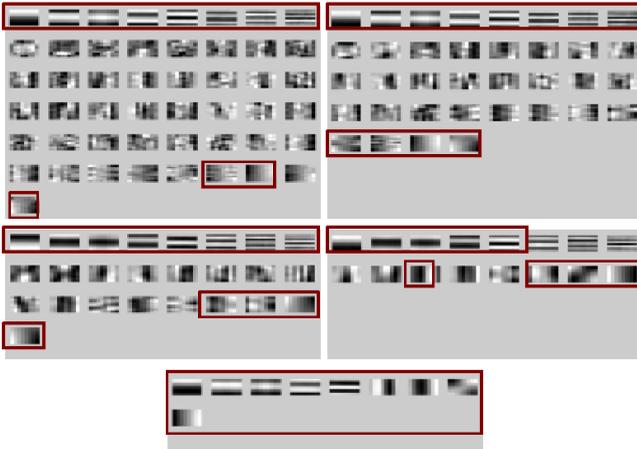


Fig. 4. Results of applying AMUSE with previous PCA to the original signals. Left superior part: reduction of dimensions to 49; right superior part: reduction of dimensions to 36; left central part: reduction of dimensions to 25; right central part: reduction of dimensions to 16; and inferior part: reduction of dimensions to 9.

4 Conclusion

In this article we have shown an analysis of the ICA algorithm AMUSE in digital images processing with noise. ICA has shown properties that allow to have a good model of the characteristics of the receivers of the cortical neurons in the human visual system. Here we have demonstrated the advantages of the ICA algorithm AMUSE that should allow investigators to choose the best algorithm according to the necessities and objectives that they have to consider.

Acknowledgements

The authors would like to thank Dr. Kurt Stadthanner from the University of Regensburg for his help with the manuscript. This work was supported in part by the project TEC 2004-0696 (SESIBONN).

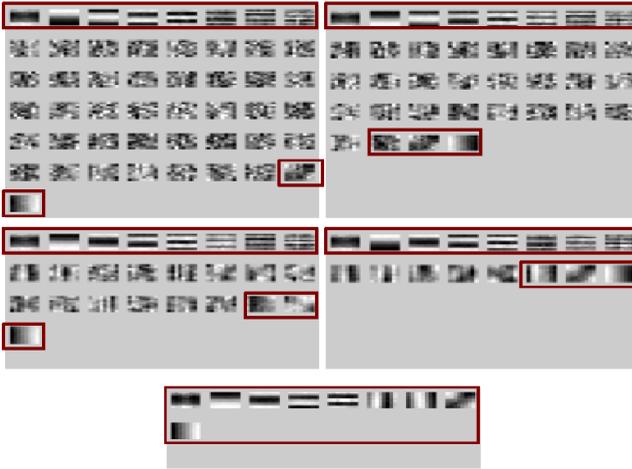


Fig. 5. Results of applying AMUSE with previous PCA to signals with gaussian noise. Left superior part: reduction of dimensions to 49; right superior part: reduction of dimensions to 36; left central part: reduction of dimensions to 25; right central part: reduction of dimensions to 16; and inferior part: reduction of dimensions to 9.

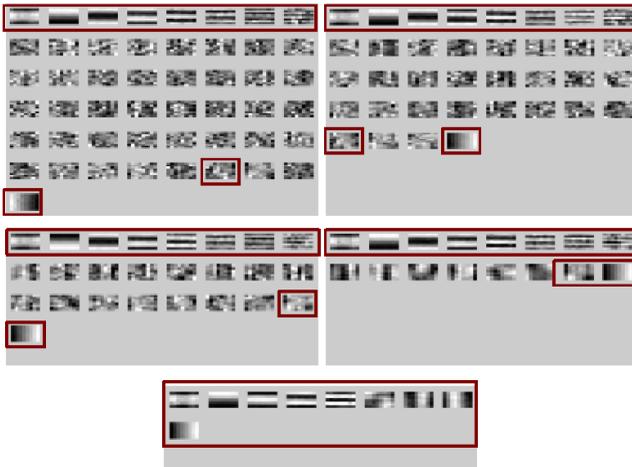


Fig. 6. Results of applying algorithm AMUSE with previous PCA to signals with multiplicative noise. Left superior part: reduction of dimensions to 49; right superior part: reduction of dimensions to 36; left central part: reduction of dimensions to 25; right central part: reduction of dimensions to 16; and inferior part: reduction of dimensions to 9.

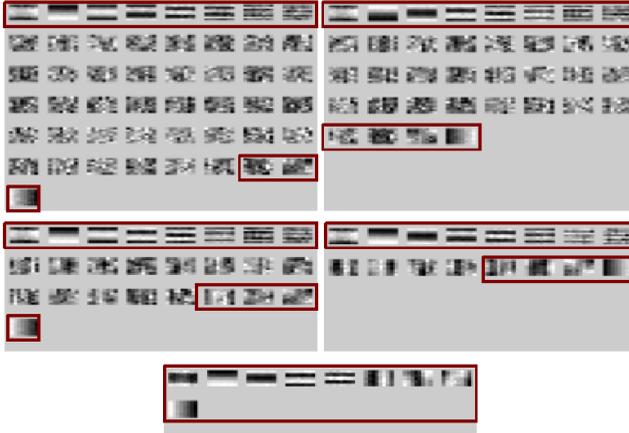


Fig. 7. Results of applying algorithm AMUSE with previous PCA to signals with salt and pepper noise. Left superior part: reduction of dimensions to 49; right superior part: reduction of dimensions to 36; left central part: reduction of dimensions to 25; right central part: reduction of dimensions to 16; and inferior part: reduction of dimensions to 9.

References

1. Hyvärinen, A.: Sparse code shrinkage: Denoising of nongaussian data by maximum likelihood estimation. In: *Neural Computation*. (1999) 1739–1768
2. Hyvärinen, A., Hoyer, P., Oja, E.: Imagen denoising by sparse code shrinkage. In: *Intelligent Signal Processing*. (2001)
3. Pajares Martin Sanz, G., De la Cruz García, J.: *Visión por computador. Imágenes digitales y aplicaciones*. RA-MA Editorial Madrid. (2001)
4. Vhalupa, J.S.: *The Visual Neurosciences*. Werner editors. MIT Press (2003)
5. Hyvärinen, A., Karhunen, J., Oja, E.: *Independent Component Analysis*. Wiley Inter-science (2001)
6. Olshausen, B.A., Field, D.J.: Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *nature*. In: *Vision Research*. (1996) 381:607–609
7. Olshausen, B.A., Field, D.J.: Sparse coding with an overcomplete basis set: A strategy employed by v1? In: *Vision Research*. (1997) 37:3311–3325
8. Bell, A., Sejnowski, T.: The independent component of natural scenes are edge filters. In: *Vision Research*. (1997) 3327–3338
9. Héroult, J., Jutten, C., Ans, B.: Detection de grandeurs primitives dans un message composite par une architecture de calcul neuromimetique en apprentissage non supervise. In: *X Colloque GRETSI*. (1985) 1017–1022
10. Jutten, C., Héroult, J.: Blind separation of sources, part i: An adaptive algorithm based on neuromimetic architecture. In: *Signal Processing*. (1991) 24:1–10
11. Comon, P.: Independent component analysis - a new concept. In: *Signal Processing*. (1994) 36:287–314
12. Tong, L., Soon, V., Huang, Y., Liu, R.: Amuse: a new blind identification algorithm. In: *Circuits and Systems, IEEE International Symposium on*. Volume vol.3. (1990) 1784–1787

Fuzzy Clustering Methods in Multispectral Satellite Image Segmentation

Rauf Kh. Sadykhov¹, Andrey V. Dorogush¹ and Leonid P. Podenok²

¹ Computer Systems Department, Belarusian State
University of Informatics and Radioelectronics
6 P. Brovka st, Minsk, Belarus

rsadykhov@bsuir.unibel.by, dorogush@lsi-bas-net.by
<http://www.bsuir.unibel.by/index.jsp?resID=100229&lang=en>

² Laboratory of System Identification, United Institute of
Informatics Problems, National Academy of Sciences of Belarus
6 Surganov st, Minsk, Belarus

podenok@lsi-bas-net.by
<http://uiip.bas-net.by/index-eng.html>

Abstract. Segmentation method for subject processing the multispectral satellite images based on fuzzy clustering and preliminary non-linear filtering is represented. Three fuzzy clustering algorithms, namely Fuzzy C-means, Gustafson-Kessel, and Gath-Geva have been utilized. The experimental results obtained using these algorithms with and without preliminary nonlinear filtering to segment multispectral Landsat images have approved that segmentation based on fuzzy clustering provides good-looking discrimination of different land cover types. Implementations of Fuzzy C-means, Gustafson-Kessel, and Gath-Geva algorithms have got linear computational complexity depending on initial cluster amount and image size for single iteration step. They assume internal parallel implementation. The preliminary processing of source channels with nonlinear filtering provides more clear cluster discrimination and has as a consequence more clear segment outlining.

1 Introduction

It is known that forests and wetland are the main factors preventing the decline in biodiversity on the Earth in aggressive conditions of human activity. The main problem is agricultural expansion and deforestation. Deforestation is the consequence of two main reasons – agricultural expansion and accidental events. But forestry and agriculture are inseparable and condemned to work hand in hand. Significant part of forest is damaged by fire, pests, irrational agricultural politics leading to change of ground water level and as result leads to sickness and wreck. At now it is possible to discriminate forest areas on early stage of damaging using multispectral images of high spatial resolution received from satellites. That technology started about 40 years ago to monitor Earth surface at now is the effective instrument of ecological and agricultural monitoring such the regions as forests and wetland and preventing any accidents. Multispectral satellite images are able to bring us information in both visible and invisible spectral bands about vegetation, water temperature and land cover.

Multi-dimensional cluster analysis and segmentation are base procedures in thematic processing the multispectral images received from remote sensing satellites. There are lot of clustering and segmentation methods which have different benefits and imperfections. The special class of such the methods is represented by fuzzy clustering ones.

2 Method Description

Three clustering algorithms based on fuzzy methods were developed and utilized as part of segmentation software. There are fuzzy c-means[1] (FCM) and its variants – Gustafson-Kessel clustering algorithm [2, 3] and Gath-Geva one [4].

FCM is a method of data clustering which allows one data objects to be a member of two or more clusters. This method developed by [5] and improved by [6] is based on minimization of the following objective function:

$$J = \sum_{i=1}^N \sum_{j=1}^C \mu_{ij}^m \|x_i - s_j\|, \quad (1)$$

where m is real number ≥ 0 , μ_{ij} – degree of membership of x_i in cluster s_j , x_i is the multi-dimensional data object, s_j is the multi-dimensional center of the cluster, and $\|\cdot\|$ is any norm expressing the similarity between any measured data and the cluster center.

Fuzzy clustering is carried out through an iterative optimization of the J_m with the update of membership matrix μ_{ij} and the cluster centers s_j using following algorithm:

1. The data set $X = (x_j) = (x_{j1}, x_{j2}, \dots, x_{jp})^T$ given. We choose the number of clusters $1 < c < n$, where p – dimension of data set.
2. Initialize the partition matrix μ_{ij} using random number generator in range $[0, 1]$:

$$\sum_{i=1}^c \mu_{ij} = 1, \quad (2)$$

3. Calculate the cluster centers $s_i = (s_{i1}, s_{i2}, \dots, s_{ip})^T$ ($i = 1, 2, \dots, c$):

$$s_i = \frac{\sum_{j=1}^n \mu_{ij}^2 x_j}{\sum_{j=1}^n \mu_{ij}^2}, \quad (3)$$

4. Calculate the error:

$$E = \sum_{i=1}^c \sum_{j=1}^n \mu_{ij}^2 (s_i - x_j)^T (s_i - x_j). \quad (4)$$

If error is small enough, then stop iterations, else go to item 5.

5. Calculate the "new" partition matrix μ_{ij} :

$$\mu_{ij} = \left(\sum_{m=1}^n d_{ij}^2 / d_{mj}^2 \right)^{-\frac{1}{2}} \quad (5)$$

and then go to step 3.

where d_{mj}^2 is similarity measure $\| \cdot \|$.

Fuzzy c-means, Gustafson-Kessel, and Gath-Geva clustering algorithms are distinguished in the definition of distance function between the objects to be classified: Fuzzy C-means:

$$d_{ij}^2 = (s_i - x_j)^T (s_i - x_j), \quad (6)$$

Simple euclidean distance provides hyper-spherical form of clusters.

Gustafson-Kessel:

$$d_{ij}^2 = (s_i - x_j)^T (\det(F_i)^{\frac{1}{2}} F_i^{-1}) (s_i - x_j), \quad (7)$$

Gath-Geva:

$$d_{ij}^2 = \frac{\det(F_i)^{\frac{1}{2}}}{\alpha_i} \exp\left(\frac{1}{2}(s_i - x_j)^T (F_i^{-1})(s_i - x_j)\right), \quad (8)$$

where F_i is covariance matrix, that provides ellipsoidal form of clusters and more comprehensive partitioning the multi-dimensional data, α_i is calculated in following manner:

$$\alpha_i = \frac{1}{n} \sum_{j=1}^n \mu_{ij}. \quad (9)$$

Covariance matrix is used to form non-spherical clusters which are more suitable for multi-dimensional data partitioning. That also leads to visible difference in algorithm convergence, effectivity and performance of processing the multispectral images.

Distance function for Gustafson-Kessel and Gath-Geva algorithms uses covariance matrix to take into account different metrics (scales) in different dimensions. The large images to be processed might be the reason of weak covariance matrix conditionality that results in heavy losses of aptitude for discrimination for Gustafson-Kessel and overflows for Gath-Geva when fixed digit capacity arithmetics based on atomic numerical types is used. To keep of that problems standard normalization methods and arbitrary (multi-precision) arithmetics are the sufficient efforts. When d_{ij}^2 becomes small the overflow may occur. To prevent that case the distant function is confined from below by some smallest value d_{min} .

Due to source signal noise the some spatial segment granularity occurs. To reduce that granularity the weak nonlinear filtering using algorithm [7] was applied to source channels. That algorithm may be used both for edges extraction and for nonlinear filtering. It does not lower the sharpness of transitions of channel brightness the boundaries of cover types remain clear. As a result the boundaries of spatial segments after clustering process also remain legible. Algorithm works with values of brightness and pixels' coordinates simultaneously. To form the homogeneous areas or search the edges on the gray-scale image a round mask is used. Usually the mask's radius is 3.4 pixels which

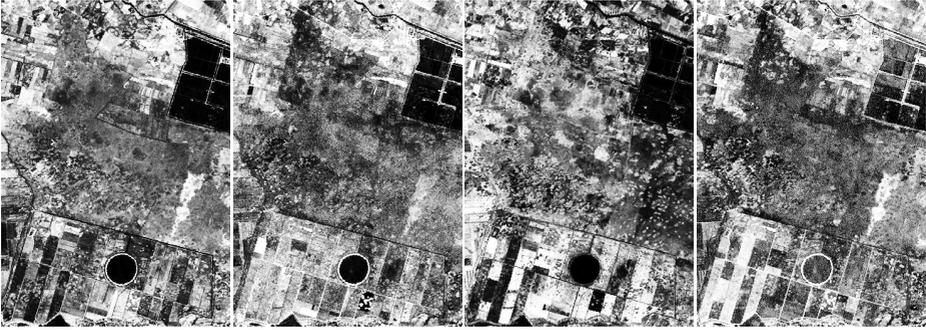


Fig. 1. Channels 2-5 of multispectral Landsat image to be processed with fuzzy clustering algorithm.

gives mask of 37 pixels size. The mask is placed at each point of the image and the brightness of each pixel of the mask is compared with the brightness of the mask central pixel.

$$c(\mathbf{r}, \mathbf{r}_0) = \begin{cases} 1, & I(\mathbf{r}) - I(\mathbf{r}_0) \leq t \\ 0, & I(\mathbf{r}) - I(\mathbf{r}_0) > t, \end{cases} \quad (10)$$

where $I(\mathbf{r}_0)$ – the brightness value of the mask's center, $I(\mathbf{r})$ – the brightness value of the mask pixel, t specific threshold.

The result of the comparisons is sum:

$$n(\mathbf{r}_0) = \sum_{\mathbf{r}} c(\mathbf{r}, \mathbf{r}_0). \quad (11)$$

where n is the quantity of pixels in the USAN (Univalue Segment Assimilating Nucleus). Then that sum should be minimized, so the algorithm is called SUSAN (Smallest USAN). Parameter t means the maximum of ignored noise. Then n is compared with it's thresholding value g , which is $3n_{max}/4$, where n_{max} – maximum value which could be assigned to n .

3 Experimental Results

To test the fuzzy clustering algorithms the multispectral Landsat images have been used. Fig. 1 shows channels 2-5 which were chosen for processing. Landsat channels shown on fig.1 have different dispersion and were equalized using `netpbm` utilities to make them visible more clearly. The channel data to be fed into clustering software remain untouched because of clustering algorithm takes in to account real distribution of channel signal level on one's own.

Raw output data of developed segmentation software is 2-d vector field over R^N , where N is a number of clusters. Each component of that field is strictly increasing function of probability for pixel to be member of one of N clusters.

On presented images one can see some well discriminated land cover types. There are open water, forests, wetlands, bushes and agriculture areas. Results of segmentation on 15 clusters using fuzzy C-means without any preliminary filtering show that there

also are fuzzy boundaries of segments on the cluster map going on to sand in some places. To diminish such the phenomena one can use filter which will smooth slightly changing areas rather blur clearly discriminated land cover type edges. Most appropriate filter having got such the behavior is the non-linear one described in [7]. The fig. 2 demonstrates smoothing properties of that filter when brightness threshold was choosed be equal to 7, 15, and 20. Besides the brightness threshold parameter the filter has spatial one – distance threshold which was set into default value corresponding to radius equal to about 3.4 and was not changed across experiments. As one can see, smooth-

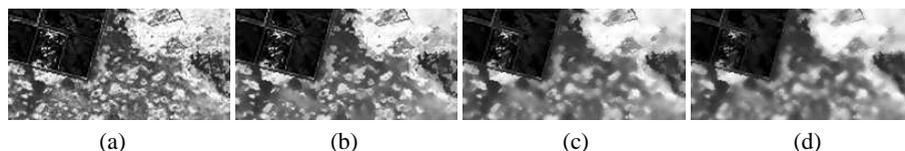


Fig. 2. Nonlinear filtering samples: (a) original image, (b) brightness threshold = 7, (c) brightness threshold = 15, (d) brightness threshold = 20.

ing property of that filter increases with brightness threshold grows. At the same time some part of edges remains sharp. Amount of sharp edges and consequently cluster granularity depend on brightness threshold. Thus we get additional degree of freedom in segmentation process control that could dramatically improve the segmentation results. Nonlinear filtering lowers intraclass covariance but practically doesn't touches

interclass ones, so cluster discrimination doesn't suffer. Smoothing the channels before segmentation allows to eliminate fine granularity of output segment map and to merge the small cluster with large ones. In reality this means disappearance of some real small objects, for example, bushes in wetland areas. Nevertheless, sometimes it is necessary to get generalized segment map. The SUSAN filter allows to fulfil nonlinear processing varyng the threshold in wide region achieveing the smoothing results from negligible small up to very strong.

Fig. 3-6 demonstrate examples of segmentation using non-linear filtering with different brightness threshold for Gustafson-Kessel and Gath-Geva clustering algorithms. The cluster number was choosen taking into account real diversity of land covers for territory being investigated. Actually a lot of experiments with different cluster number were fulfilled. As soon as segmentation become to look stable cluster number was stated.

Obtained results have transformed into scalar field using simple maximal probability solver to get the possibility of visual evaluation and have presented as colour map of spatial partitions. As a result every pixel became a member of one cluster. All pixels of same colour are members of same cluster. No any intelligent algorithm was used to colourize segment map so the same areas on different pictures have different colours.

As one can see the growth of the threshold leads to raising the segment area when target number of clusters is fixed. However it should be pointed that general structure of segment arrangement and its structure in relation with land cover types is not significantly changed.

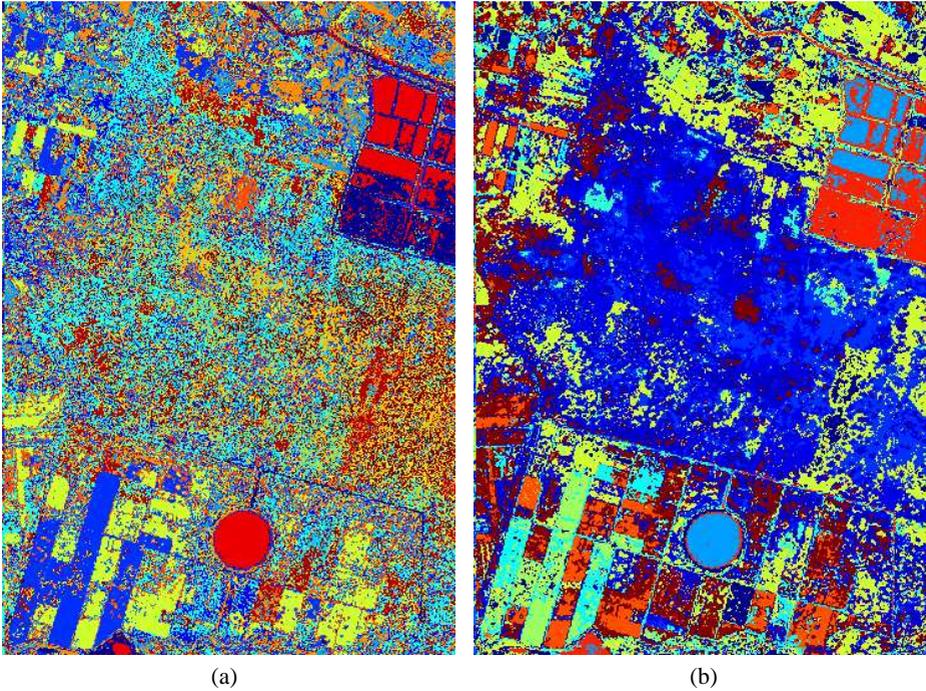


Fig. 3. Segmentation results using Gustafson-Kessel (a) and Gath-Geva (b) algorithms at 20 clusters without any filtering.

4 Conclusion

Segmentation results of multispectral Landsat images obtained using fuzzy clustering methods such as Fuzzy C-means, Gustafson-Kessel, and Gath-Geva with and without preliminary nonlinear filtering testify that segmentation using fuzzy clustering methods provides good-looking discrimination of land cover types that occurs in such complex cases as wetland, water-meadow, and bush areas. The discrimination quality of segmented images was tested and approved by land-improvement specialists using data of land-based expedition. The non-linear filtering with guided smoothing is the convenient instrument of preliminary processing for semi-automatic segmentation of complex land covers.

Implementations of Fuzzy C-means and Gustafson-Kessel algorithms have got linear computational complexity depending on initial cluster amount and image size for single iteration step. All the algorithms assume internal parallel implementation for MPP computer. The preliminary processing of source channels with nonlinear filter provides more clear cluster discrimination and has as a consequence more clear segment outlining and provides operated generalization of output segment maps.

Really, segmentation software returns the results as 2-d vector field v_i , which is the strictly increasing function of probability p_i that pixel is a member of i -th cluster. When using more complex tool than simple maximal probability solver, for example, maximal

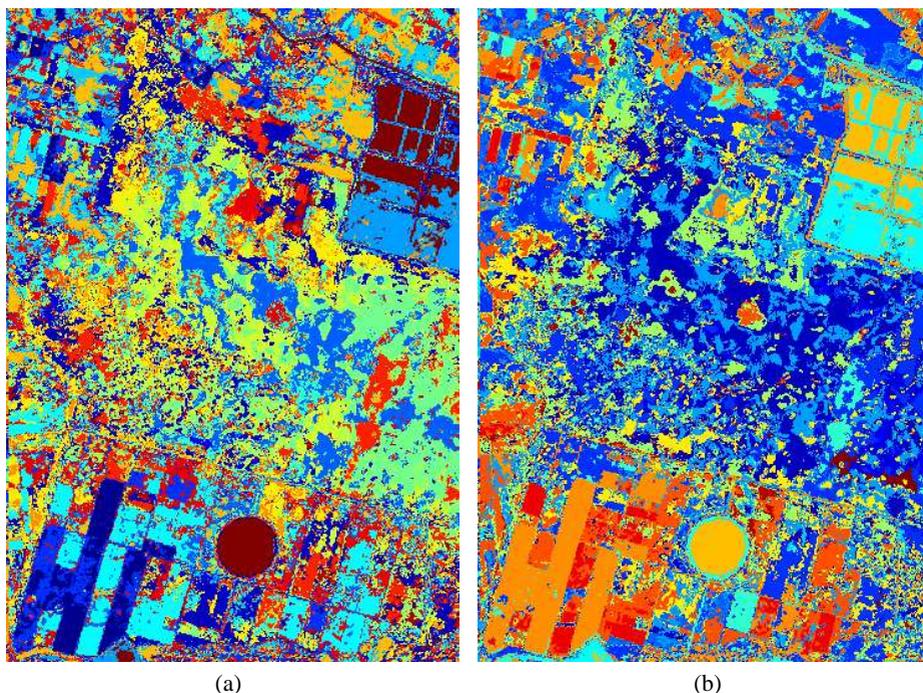


Fig. 4. Segmentation results using Gustafson-Kessel (a) and Gath-Geva (b) algorithms at 20 clusters using filtering with threshold = 7.

likelihood one, it is possible to significantly improve the results of segmentation and to avoid large part of manual processing the multispectral data.

References

1. Hoppner, F., Klawonn F., Kruse, R., Runkler, T.: Fuzzy Cluster Analysis. Wiley, Chichester (1999).
2. Gustafson, D. E., Kessel, W. C.: Fuzzy clustering with fuzzy covariance matrix. In Proceedings of the IEEE CDC, INSTICC Press, San Diego (1979) 761–766
3. Babuska, R., van der Veen, P. J., Kaymak, U.: Improved covariance estimation for Gustafson-Kessel clustering. IEEE International Conference on Fuzzy Systems (2002) 1081–1085”
4. Gath, I., Geva, A. B.: Unsupervised optimal fuzzy clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence (1989) 7:773–781
5. Dunn, J. C.: A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. Journal of Cybernetics (1973) 3: 32-57
6. Bezdek, J. C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
7. Smith, S. M., Brady, J. M.: SUSAN – a new approach to low level image processing. International Journal of Computer Vision May (1997) 23(1):45-78

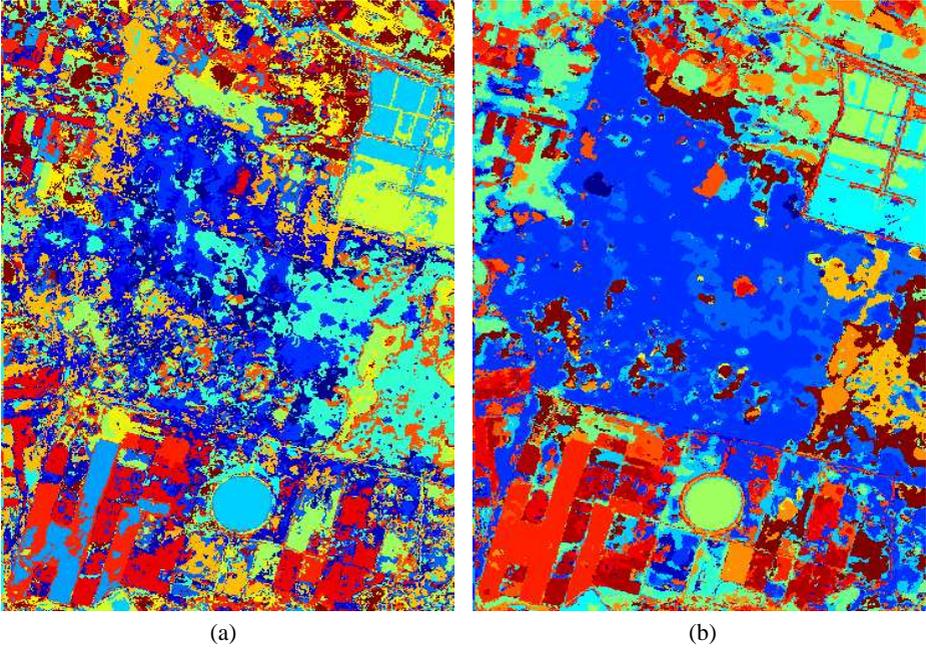


Fig. 5. Segmentation results using Gustafson-Kessel (a) and Gath-Geva (b) algorithms at 20 clusters using filtering with threshold = 15.

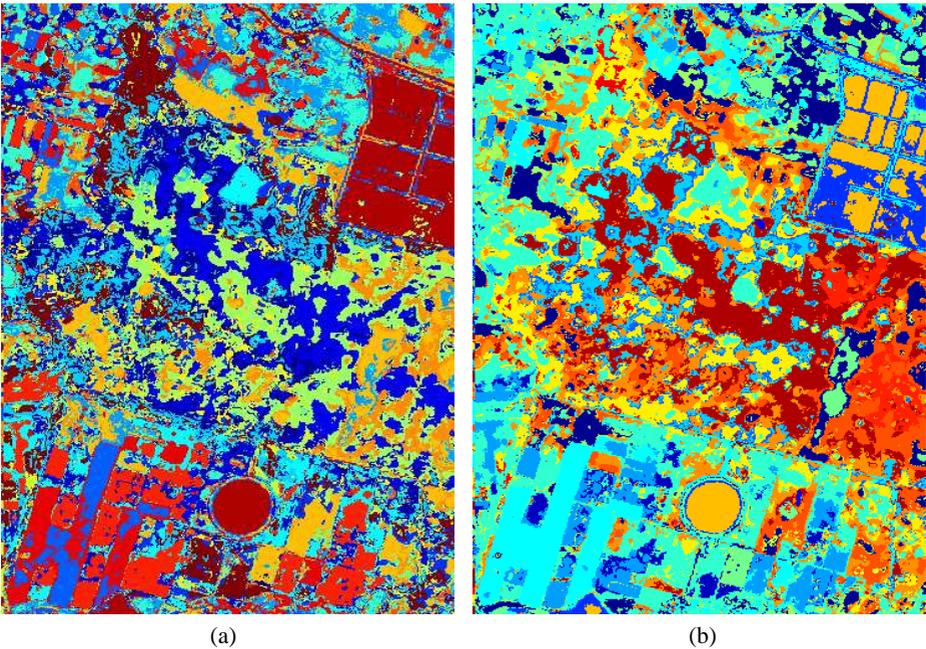


Fig. 6. Segmentation results using Gustafson-Kessel (a) and Gath-Geva (b) algorithms at 20 clusters using filtering with threshold = 20.

Automatic Classification of Spinal Deformity by using Four Symmetrical Features on the Moire Images

Hyoungeop Kim¹, Satoshi Nakano¹, Joo Kooi Tan¹, Seiji Ishikawa¹
Yoshinori Otsuka², Hisashi Shimizu³ and Takashi Shinomiya⁴

¹ Department of Control Engineering, Kyushu Institute of Technology
1-1, Sensui-cho, Tobata, Kitakyushu, Japan
kim@cntl.kyutech.ac.jp

² National Sanatorium Chiba Higashi Hospital, Japan

³ Chiba Foundation for Health Promotion & Disease Prevention, Japan

⁴ Nikon Co. LTD., Japan

Abstract. Spinal deformity is a disease mainly suffered by teenagers during their growth stage particularly from element school to middle school. There are many different causes of abnormal spinal curves, but all of them are unknown. The most common type is termed “idiopathic” that show 80% of the spinal deformity. Spinal deformity is a serious disease, mainly suffered by teenagers, especially girl’s student, during their growth stage. To find the spinal deformity in early stage, orthopedists have traditionally performed on children a painless examination called a forward bending test in mass screening of school. But this test is neither objective nor reproductive, and the inspection takes much time when applied to medical examination in schools. To solve this problem, a moire method has been proposed which takes moire topographic images of human subject backs and checks symmetry/asymmetry of the moire patterns in a two-dimensional way. In this paper, we propose a method for automatic classification of spinal deformity from moire topographic images by extracting four symmetrical features of the left-hand and right-hand side on the moire image. Feature of asymmetry degrees are applied to train employing the classifier such as Artificial Neural Network, Support Vector Machine, Self-Organization Map and AdaBoost.

1 Introduction

Spinal deformity is one of the serious diseases, mainly suffered by teenagers. It tends to run in families and is more common in females than males during their growth stage. There are many different causes of spinal deformity such as congenital, kyphosis (curvature of the spine with the convexity pointing toward the back), but in the vast majority of cases there is no known cause. Although the spine does curve from front to back side it should not curve lateral. A side-to-side called scoliosis and it may take the shape of an ‘S’ or ‘C’ character. The difficulty because of not accompanied by the subjective symptom such as pains the early stage detect and the early treatment becomes a problem. When one suffers from a spinal deformity, in severe

case, it is associated with pain and it requires surgical treatment. The treatment of spinal deformity depends on the location and degree of curvature. Slight curves usually require no treatment, but as the curve progresses the treatment is required because the size of chest cavity diminish, it causes pain and decrease in lung.

To find the spinal deformity in early stage, orthopedists have traditionally performed a painless examination which called the forward bending test in mass screening of school. In the forward bending test, mainly medical doctor checks 5 points such as rib hump, lumbar hump, and asymmetric degree on the shoulder and waist line. But this test is neither objective nor reproducible, and the inspection takes much time when applied to medical examination in school screening. To solve such various problems, a moire method [1-2] has been proposed which takes moire topographic images of human subject backs and checks symmetry/asymmetry of the moire patterns in a two-dimensional way. The moire topographic image represented stripe pattern as one of the three dimension information. Moire stripes appear as symmetry the subject is classified as normal.

By using the moire image, the diagnosis efficiency of spinal deformity in the mass screening improved. However, the burden of the doctor who diagnoses a large amount of moire image is still remained. Then, the necessity of the image diagnosis support by using a computer is requested from the medical site. To detect the spinal deformity, some algorithms are proposed [3-4]. In this paper, we propose a technique for automatic classification of spinal deformity from moire topographic images by extracting four symmetrical features of the left-hand and right-hand side on the moire image. In the first step, once the original moire images is fed into computer, the middle line of the subject's back is extracted on the moire image employing the approximate symmetry analysis[5]. Regions of interest (ROIs) are automatically selected on the moire image from its upper part to the lower part and the middle line of the subject's back. Then the four asymmetry degrees are calculated from obtained ROIs. Numerical representation of the degree of asymmetry, displacement of local centroids and difference of gray value, are calculated between the right-hand side and the left-hand side regions of the moire images with respect to the extracted middle line. Feature of four asymmetry degrees (mean value and standard deviation from the each displacement) from the right-hand side and left-hand side rectangle areas apply to train the Artificial Neural Network (ANN), Support Vector Machine (SVM), Self-Organization Map (SOM) and AdaBoost.

2 Extraction of the Middle Line

Moire photography uses light projected through a grid and then photographed to record the 3-D shape of the subject's back. Generally, the moire stripes show symmetric patterns on the normal subject's backs. But when one becomes spinal deformity, an asymmetric moire pattern appears on the moire image of the subject's back. In the diagnostic of imaging by using the moire method, asymmetry degree are evaluated on the moire images, so it is effective to make the asymmetry degree on the moire image in the visual screening.

To analyze the asymmetric of moire pattern, the middle line is extracted based on approximately symmetry analysis technique [5]. The approximate symmetric axis can be found by superposing the original and the reflected original image (mirror image). The best position of the superposing is determined, by evaluating the difference image which is overlapped the original and the mirror image. We adjusted to the position in which the difference of the density of a pixel values are minimized. The approximate symmetric axis is represented by the perpendicular bisector of the center of gravity of the original and the mirror image.

We assume an original moire image is $f(x,y), (x,y) \in R$, and its reflected image is represented by $f'(x,y), (x,y) \in R'$. The $f'(x,y)$ is superposed onto the $f(x,y)$ by parallel translation $c=(c_x, c_y)$, T is a rotation transform and rotation θ to find the best match in eq.(1). In this paper, we assume that $\theta=0$ in eq.(2), because the moire images are captured normally straight using position-supporter so that their middle lines remain vertical.

$$D_{axis} = \min_T \sum_{(x,y) \in R \cup R'} |f(x,y) - Tf'(x,y)| \quad (1)$$

$$T = \begin{pmatrix} \cos \theta & \sin \theta & c_x \\ -\sin \theta & \cos \theta & c_y \\ 0 & 0 & 1 \end{pmatrix} \quad (2)$$

3 Extraction the Four Asymmetrical Features on the ROIs

The ROIs are selected by using pre-processing technique, the asymmetrical features are calculated by the following way.

Within the region R and at a certain position $y=j$, two rectangle areas are defined, as shown in Fig.1, at symmetric locations with respect to the middle line $x=m$. The width R_x of the rectangle area is defined by,

$$R_x = \min(m-l, r-m). \quad (3)$$

Here m is the middle line which is extracted above mentioned, l is minimum frequency of the left-hand side r is minimum frequency of the right-hand side on the histogram. On the other hand, height of the area is defined empirically.

Let us denote the rectangle areas of the left-hand side and right-hand side at $y=i$ by A_i^l and A_i^r , respectively. Here $i=1,2,\dots,N$. The centroids of A_i^l and A_i^r are denoted by $G_l(x_l, y_l)$ and $G_r(x_r, y_r)$, respectively. The centroid $G_l(x_l, y_l)$ is reflected with respect to the middle line $x=m$ into the region A_i^r and denoted by $G_l^*(x_l^*, y_l^*)$. The distance E between $G_l^*(x_l^*, y_l^*)$ and $G_r(x_r, y_r)$ is calculated by,

$$E = \sqrt{(x_l^* - x_r)^2 + (y_l^* - y_r)^2} \quad (4)$$

The mean μ_E and standard deviation σ_E of the values E ($i=1,2,\dots,N$) are employed as the features representing the degree of asymmetry of the moire image in calculation rectangle area. The expressions are shown as follows.

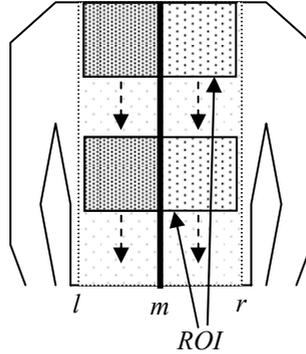


Fig. 1. Rectangle areas in the region of interest.

$$\left\{ \begin{array}{l} \mu_E = \frac{1}{N} \sum_{i=1}^N E \\ \sigma_E = \sqrt{\frac{1}{N} \sum_{i=1}^N (E - \mu_E)^2} \end{array} \right. \quad (5)$$

Furthermore, in the same rectangle area, the difference of gray value D is calculated by,

$$D = |r_d - l_d| \quad (6)$$

Here, r_d and l_d are shown the mean value of the gray value on the right-hand and left-hand side in the ROIs, respectively. The mean μ_D and standard deviation σ_D of the difference of gray values D ($i=1,2,\dots,N$) are employed as the features representing the degree of asymmetry of the moiré image in calculation rectangle area. The expressions are shown as follows.

$$\left\{ \begin{array}{l} \mu_D = \frac{1}{N} \sum_{i=1}^N D \\ \sigma_D = \sqrt{\frac{1}{N} \sum_{i=1}^N (D - \mu_D)^2} \end{array} \right. \quad (7)$$

4 Classification Methods

The mean value and the standard deviation of the difference of the center of gravity and difference of gray value on the right-hand and left-hand side area are obtained as the feature, respectively. To classify the unknown moiré image, we have tried the ANN, SVM, SOM and AdaBoost techniques employing four asymmetrical features.

ANN is used a useful technique for the pattern classification. This technique provided a method for the automatic spinal deformity. It is necessary for input layers,

which extracted numerical feature. To classifying the unknown moire image, the four features (mean values and standard deviation in eq.(5), (7)) are used for training by using the back propagation in ANN. Our ANN is consist of three layers, which include four inputs neurons, three hidden neurons and two output neurons for training. Finally, unknown moire images are discriminated as normal or abnormal case automatically.

A SVM [6, 7] is a supervised learning technique from the field of machine learning applicable to both classification and regression. SVM is a set of related supervised learning methods used for classification. It is an optimization algorithm for the problem of pattern recognition. Some free software also provided methods for assessing the generalization performance efficiently. It was worked out for linear two-class classification with margin, which has the minimal distance from the separating hyperplane to the closest data points. SVM learning machine seeks for an optimal separating hyper plane, where the margin is maximal. In this method, to classify the unknown moire images, we implement the SVM technique employing four feature vectors from the left-hand side and right-hand side of rectangle areas (in eq.(5) and eq.(7)).

SOM [8] is a data visualization technique invented by T. Kohonen which reduces the dimensions of data through the use of self-organizing neural networks. In this study, we applied our method to the SOM for clustering the normal and abnormal moire image.

AdaBoost [9] is the useful technique of the Boosting technique. Boosting makes a learning machine different as the weight of the exercise is changed one after another, the technique which composes the learning machine that these are combined and accuracy is high. In AdaBoost when the weight of the learning machine is updated, weight to the training sample misclassified with the learning machine increases, and weight to the training sample correctly classified decreases. Therefore, it might become difficult to see the whole image because data with a difficult distinction is emphatically learned.

5 Experimental Results

Experiment was done employing 1200 real moire images which is 600 of abnormal and normal, respectively. The employed moire images are separated into two groups such as training and test data sets. As a training data for this study, we have selected randomly 400 (200 normal and abnormal cases, respectively) moire images which is called G_1 , G_2 , and G_3 . The leave-one out method is applied onto three data groups, and the average recognition rate is calculated. The leave-one out is a method of applying the obtained criteria to the data group of the remainder for two data groups, doing the evaluation to which data is not biased.

The employed moire topographic image size is 256X256 pixels with 256 gray levels. Fig.2 illustrates experimental results. In Fig.2, (a) shows a normal moire image and (b) shows an abnormal moire image. Table 1 shows obtained classification rates. In the table, G_i ($i=1,2,3$) shows data sets, “*Normal*” shows classification rates which normal cases were classified correctly, and “*Abnormal*” shows classification rates

which abnormal cases were classified correctly. Finally, “Average” shows the average classification rate obtained from each data group, “Ave.” shows the entire average classification rate. That is, the paragraph of G_1 shows the identification rate when G_2 and G_3 are learned as learning data, and the result of obtaining is applied to G_1 . As a result, on the total average, classification rate of 85.2%, 85.3%, 71.8%, and 85.6% were achieved in the ANN, SVM, SOM, and AdaBoost, respectively.

6 Discussion and Conclusions

In this paper, we proposed a new automatic classification method for the spinal deformity detection by using ANN, SVM, SOM, and AdaBoost method which is extracted asymmetry degree. The middle line of the subject’s back is extracted on moire image employing the approximate symmetry analysis, and ROIs are automatically selected, then the asymmetry degree is calculated. Four asymmetry degrees from the right-hand and left-hand side rectangle areas which is selected as ROIs apply to train the ANN, SVM, SOM, and AdaBoost. The total average shows the classification rate of 85.2%, 85.3%, 71.8%, and 85.6% in the ANN, SVM, SOM, and AdaBoost respectively in the experiment employing 1200 moire image. In the experimental results, the average classification rate of spinal deformity by ANN and AdaBoost was slightly higher than the other classifier.

Fig.3 illustrates examples of misclassification result. In Fig.3, a normal case is classified into abnormal in (a), whereas an abnormal case is classified into normal in (b). In figure 3, sunburn trace appears on the waist part in (a). In Fig. 3 (b), gray values subtly differ in the vicinity of an edge particularly on the shoulder part. All of the misclassified normal cases are found asymmetry of moiré patterns. This is because gray values distribution in the rectangle regions unfortunately affected symmetrically when the features were calculated. To escape from this difficulty, some other asymmetry features such as asymmetric of shoulders line or asymmetric of angle on a waist line might be taken into account in conjunction with it. These issues remain for further study.

In the experimental results, the classification rates which normal cases were classified correctly are higher than the classification rates which abnormal cases were classified correctly. Generally, medical doctor checks the symmetric shape of right-hand and left-hand side such as waist line and shoulder line of human back. In the normal case, waist line shows almost symmetric shapes. On the other hand, in the abnormal case, asymmetric moire patterns are appeared on the waist line. To improve the classification rate in the future, we introduce a new feature such as waist line and shoulder line for the new features. That still remained as a future works.

Acknowledgements

This work was supported by a Grant-In-Aid for Scientific Research on Priority Areas (18560414) from the Ministry of Education, Culture, Sports, Science and Technology, Japan.

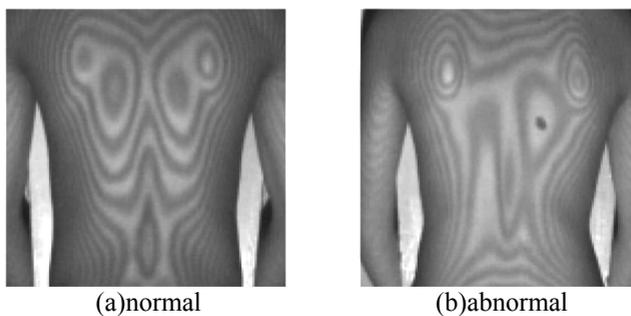


Fig. 2. Experimental results.

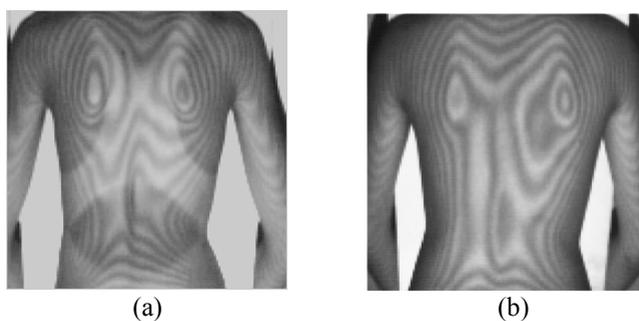


Fig. 3. Examples of misclassification: (a) Classified normal to abnormal; and (b) Classified abnormal to normal.

Table 1. Classification rates [%].

	G_1	G_2	G_3	<i>Ave.</i>
ANN				
<i>Normal</i>	75	79.5	76	75
<i>Abnormal</i>	96.5	90	94	94
<i>Average</i>	85.8	84.8	85	85.2
SVM				
<i>Normal</i>	76	78.5	73	75.8
<i>Abnormal</i>	98	91	79.5	94.8
<i>Average</i>	87	84.8	84.3	85.3
SOM				
<i>Normal</i>	68	74.5	74	72.2
<i>Abnormal</i>	75	67.5	71.5	72.9
<i>Average</i>	71.5	71	72.8	71.8
AdaBoost				
<i>Normal</i>	75.5	81	80	78.8
<i>Abnormal</i>	95.5	90	91.5	92.3
<i>Average</i>	85.5	85.5	85.8	85.6

References

1. Y. Ohtsuka, A. Shinoto, and S. Inoue, "Mass school screening for early detection of scoliosis by use of moire topography camera and low dose X-ray imageing", *Clinical Orthopaedic Surgery*, 14, 10, pp.973-984, 1979. (in Japanese).
2. H. Takasaki, "Moire topography from its birth to practical application", *Optics and Lasers in Engineering*, 3, pp.3-14, 1982.
3. Kim, H.S., Ishikawa, S., Ohtsuka, Y., Shimizu, H., Sinomiya, T., Viergever, M.A. "Automatic scoliosis detection based on local centroids evaluation on moire topographic images of human backs", *IEEE Transaction on Medical Imaging*, Vol.20, No.12, pp.1314-1320, 2001.
4. Batouche, M. "A knowledge based system for diagnosing spinal deformations Moire pattern analysis and interpretation", *International Conference of Pattern Recognition*, pp.591-594, 1992.
5. Minovic, P., Ishikawa, S., Kato, K. "Symmetry identification of a 3-D object represented by octree", *IEEE Trans. Patt. Anal. Machine Intell.*, PAMI-15, 5, 507-514, 1993.
6. V. Vapnic: *The nature of statistical learning theory*, Springer-Verlag, New York, 1995.
7. Christopher J. C. Burges "A Tutorial on Support Vector Machines for Pattern Recognition", *Data Mining and Knowledge Discovery* 2, pp.121 - 167, 1998.
8. T. Kohonen *Self-organizing maps*, Springer-Verlag New York, Inc., Secaucus, NJ, 1997.
9. Y. Freund, R. E. Schapire, "Experiments with a New Boosting Algorithm", *International Conference on Machine Learning*, pp.148-156, 1996.

Improved Neural Network-based Face Detection Method using Color Images

Yuriy Kurylyak¹, Ihor Paliy¹, Anatoly Sachenko¹, Kurosh Madani²
and Amine Chohra²

¹ Research Institute of Intelligent Computer Systems
Ternopil National Economic University
3 Peremoga Square, 46004, Ternopil, Ukraine
{as, ipl, yuk}@tanet.edu.te.ua

² Images, Signals and Intelligent Systems Laboratory (LISSI / EA 3956)
PARIS XII University, Senart-FB Institute of Technology
Av. Pierre Point, Bat. A, F-77127, Lieusaint, France
{madani, chohra}@univ-paris 12.fr

Abstract. The paper describes some face detection algorithms using skin color segmentation, Haar-like features and neural networks. The segmentation using skin color labels promising input image areas that may contain faces. The usage of Haar-like features allows fast rejection of the majority of background. Then, the ensemble of retinally connected neural networks performs the final classification of the rest image windows using improved face search strategy across scale and position. The proposed search strategy applies inverse image scale pyramid, adaptive scanning step and window acceptance to decrease the number of windows which should be processed by the classifier.

1 Introduction

Human face detection (FD) is a very important quick-developing research area which has a wide range of applications, like face recognition, video-conference, content-based image retrieval, video-surveillance, etc. FD is also a challenging task because of facial variability in scale, location, orientation and pose. Many different FD approaches have been proposed in the last years: knowledge-based, invariant feature-based, template matching, and appearance-based [1]. The earlier methods are based on human-coded rules or facial features which are invariant to pose and orientation change, difficulty handle cluttered scenes with complex background and detect a lot of false positives [1]. Some facial features like a skin color may be used to select face candidate regions which extremely reduces the search area. Then these regions may be processed by more complex and accurate classifier. The simplest skin color segmentation method is pixel-based skin color detection with explicitly defined skin cluster boundaries in some color-spaces [2]. Applying of some Haar-like features also reduces the search space [3].

More recent FD methods from appearance-based group show excellent results on benchmark test sets with variable faces in uncontrolled environment. Sung and Pog-

gio developed a distribution-based approach for FD which was the first accurate appearance-based method [4]. Training examples are gathered from creation of virtual faces and bootstrapping. Each face and non-face is normalized using masking, illumination gradient correction and histogram equalization. All training patterns are grouped into six face and six non-face clusters. Euclidean and normalized Mahalanobis distances are computed between an input image pattern and the prototype clusters. Multilayer perceptron network is applied to classify face window patterns from non-face patterns using the distances to each face and non-face cluster.

The first advanced neural network-based approach that reported results on a large and difficult dataset was by Rowley et al. [5]. It becomes de-factor the standard for evaluation with other upright frontal FD approaches. Their system incorporates face knowledge in a retinally connected neural network, looking at windows of 20x20 pixels. In their single neural network implementation, there are two copies of a hidden layer with 26 units, where 4 units look at 10x10 pixel sub-regions, 16 look at 5x5 sub-regions, and 6 look at 20x5 pixels overlapping horizontal stripes. The input window is pre-processed like in the Sung and Poggio's system [4]. The image is scanned with a moving 20x20 window at every possible position and scale with a subsampling factor of 1.2. To reduce the number of false alarms, they combine multiple neural networks with an arbitration strategy. The fast version of FD system uses extra neural network that scans an image with 30x30 pixels window and 10 pixels step for face candidates which then are passing to the verification neural network.

A new extremely fast FD algorithm is presented by Viola and Jones [3] that uses AdaBoost for selecting essential Haar-like features and the attention cascade of classifiers.

The state of the art methods [3, 5] still have some disadvantages. For example, FD system which is based on [3] misses partially-occluded or hardly shadowed faces and gives more false positive than in [5], whereas FD approach which is described in [5] is too slow for real-time video-flow processing. In our paper we propose to combine the abovementioned approaches to overcome these disadvantages by using some Haar-like features from [3] for face candidate selection and improved FD neural network-based method, adapted from [5]. We also used color segmentation preprocessing stage with image color balance enhancement, skin detection in several color-spaces and morphological operations for the FD process acceleration. After the preprocessing stages the final FD is performed using improved face search strategy across scale and position with the following key elements: inverse image scale pyramid, adaptive window scanning step and window acceptance. These improvements in search strategy allow reducing the number of handled windows especially in the case of large faces presence. Training set for neural network is formed in bootstrap manner not only for non-faces but also for faces. This provides to draw a distinction between two classes more precisely.

The rest of this paper is organized as follows: first, we describe face candidate selection algorithms which are based on skin color segmentation and Haar-like features' analyzing, in section 3 the improved neural network-based method is described in details and in the last section the conclusions and the future directions of our research are given.

2 Face Candidate Selection

2.1 Face Candidate Selection Using Skin Color Segmentation

The human skin has a characteristic color and could be easily recognized by people. Therefore, the usage of skin color (SC) information can considerably facilitate the process of faces exposure, localization and tracking [2]. Color allows fast processing of the input image and is highly robust to geometric variations of the face pattern.

SC segmentation can be based on separate pixels or on regions. In this work we use pixel segmentation, including classifier creation to separate skin-pixels from the background. The classifier creation is accomplished by determination of the metrics that measure distances between the pixel color and SC. The metrics type is defined from the SC modeling method: explicitly defined skin region (defining skin region boundaries), nonparametric skin distribution modeling (defining of the skin color distribution from training set), parametric and dynamic skin distribution modeling [2]. We use the method of explicitly defined skin region boundaries as it is simple, fast, and exact enough.

There are a few color spaces which successfully apply for segmentation tasks: RGB, nRGB, HSV, TSL, HSI, YIQ, YCbCr and other. Our experiments show that the best segmentation is provided by the combination of RGB and TSL color-spaces (Fig. 1). We use the following rule to determine the boundaries of the SC cluster in RGB color space (for each of the R, G, and B channels) [6]:

% The skin color model at uniform daylight illumination

$$\begin{aligned}
 &R > 95 \text{ and } G > 40 \text{ and } B > 20 \text{ and} \\
 &\max\{R, G, B\} - \min\{R, G, B\} > 15 \text{ and} \\
 &|R - G| > 15 \\
 &\text{and } R > G \text{ and } R > B
 \end{aligned}$$

OR

% The skin color model under flashlight lateral illumination

$$\begin{aligned}
 &R > 220 \text{ and } G > 210 \text{ and } B > 170 \text{ and} \\
 &|R - G| \leq 15 \text{ and} \\
 &R > B \text{ and } G > B
 \end{aligned}$$

The usage of the additional spaces (YCbCr, YIQ) allow to reject some more background pixels, but the speed of segmentation block executing will fall down.

Color balancing is performed before the segmentation to adjust color distribution. The segmentation is followed by the morphological operations (opening, closing, and filtration) in order to improve an image quality (Fig. 2).

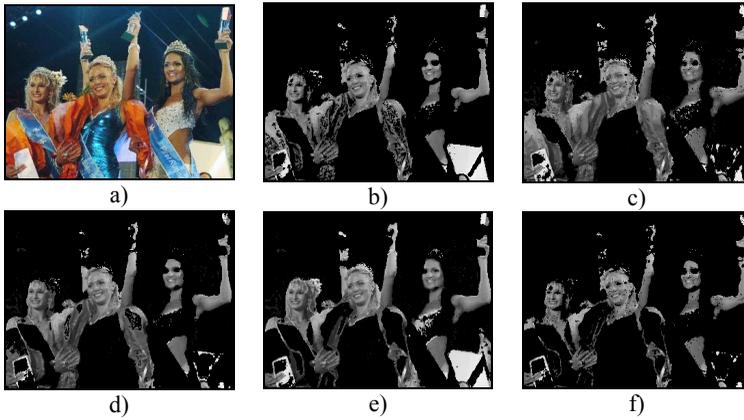


Fig. 1. Segmentation results of input image (a) using RGB (b), TSL (c), YCbCr (d), YIQ (e) color spaces and the result of their combination (f).

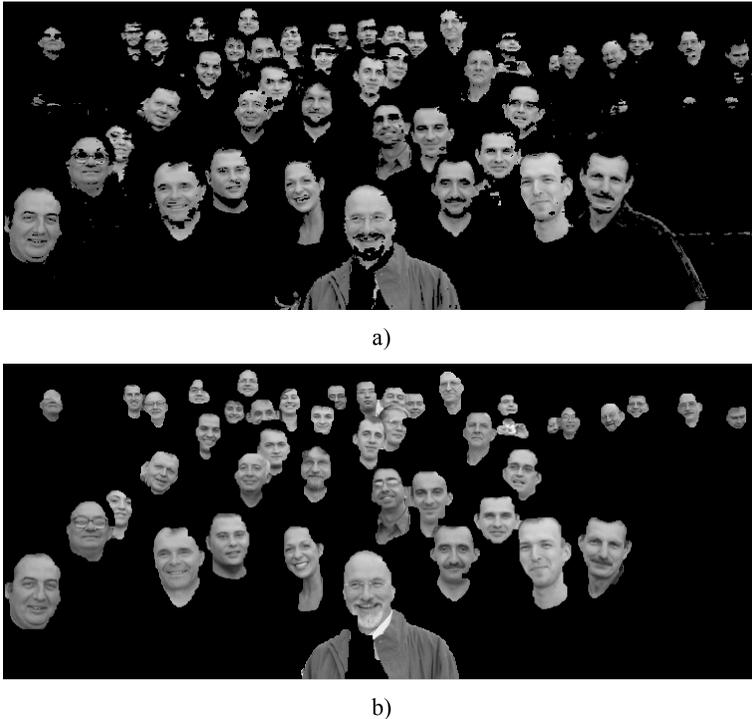


Fig. 2. Segmented image before (a) and after (b) applying of the morphological operations.

SC segmentation allows extremely reduce a face search area and speedup the whole FD process in 5-20 times depending on the input image.

2.2 Face Candidate Selection Using Haar-like Features

We use some Haar-like features, presented in [3], as a preprocessing step to reduce the face search area (Fig. 3). The size and position of these features is selected in order to provide the error less than 1% on the training set. The features also used on the training stage to reduce the number of non-face images, gathered during the bootstrapping.

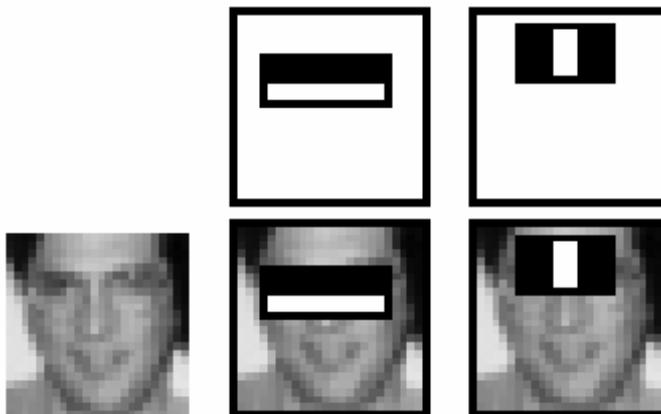


Fig. 3. First two Haar-like features [3].

In comparison with [5] the usage of these features extremely reduces the number of analyzed sub-images for the final classifier (see Section 3).

3 Improved Neural Network-Based Face Detection Method

3.1 Neural Network Active Training Algorithm

The face images for the training set, which were collected from MIT CBCL face data set [7] and Internet, were scaled and cropped to the size of 20x20 pixels. The training set was extended using virtual examples creation by randomly mirroring, rotating, scaling, translating and blurring each of the original face samples. Unlike classical virtual examples creation procedure described in [4, 5] we translate training face samples by 0.5 and 1 pixel vertically and horizontally purposely, to increase the default window scanning step to 2 pixels. We also used blurring operation to extend the training set with cinema-like faces. The total size of the training set is 3242 face images.

We used active training algorithm for retinally connected neural network [5] with a bootstrapping procedure extended on faces where masking, illumination gradient correction and histogram equalization were applying for each of the training sample. Active training algorithm consists of the following steps, adapted from [5]:

1. Create an initial training set by randomly selecting 500 face images from the whole face set and generating 500 random non-face images. Apply the preprocessing steps to each of these images.
2. Train a neural network to produce an output of 0.9 for the face examples and -0.9 for the non-face examples. The training algorithm is a scaled conjugate gradient back-propagation. If mean square error is too large, find the training sample with the biggest error and exclude it from the current training set. Go to step 2.
3. Run the system on images which contain no faces. Randomly collect 25 sub-images in which the network incorrectly identifies faces as negative examples.
4. Run the system on the whole face set. Randomly collect 25 face images in which the network incorrectly identifies non-faces as positive examples. If the number of collected images smaller than 25, randomly select the deficient images from the whole face set.
5. Apply the preprocessing steps to collected face and non-face images and add them to the current training set. Go to step 2.

Such training algorithm provides the network with relatively small representative training set (5440 images after 100 training epochs) since the network is collecting face and non-face examples itself. The testing of the trained neural network was performed on MIT CBCL face test set [7] which includes 472 face and 23573 non-face images and the average error was 1.96%.

3.2 Improved Face Search Strategy Across Pose and Scale

The classical face search strategy (FSS) across pose and scale supposes the gradual decrease of the input image with some scale coefficient and FD is performed by shifting a search window over the input image with some moving step (usually it equals to 1). Then each of the sub-images is classified to face/non-face class using a classifier [4, 5, 8]. We propose to improve the FSS using inverse image scale pyramid, adaptive window scanning step and window acceptance. These improvements allow decreasing the number of sub-images processed by classifier.

The image scale pyramid is constructed from the smallest image (usually equals to scanning window size) to its original size (Fig. 4).



Fig. 4. The image scale pyramid.

First, the neural network-based classifier looks for large faces. When the face candidate region has some number of position and scale detections this face can be accepted and its image region can be eliminated from further processing (Fig. 5). This

verification requires the on-line registration of multiple detections during the detection process unlike the off-line detection results processing used in [5].

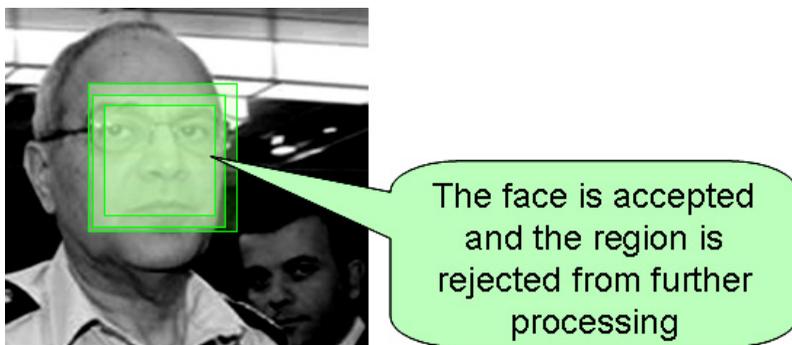


Fig. 5. Face window acceptance.

The classifier avoids analyzing of the accepted face regions using adaptive window scanning step when looking for smaller faces. The default value of adaptive step is 2 (along rows and columns) and it changes in the following cases:

- face-like region (region with a deficient number of multiple detections) is found: the step decreases to 1;
- face candidate is found: the step essentially increases one-time and then sets to its default value;
- accepted region is found: the step essentially increases one-time.

Table 1 shows considerable diminishing of the sub-images number which is analyzed by the neural network using adaptive step and Haar-like features while processing a 71x74 grayscale image (Fig. 5) (experiments are performed in Matlab environment).

Table 1. Face detection using improved face search strategy.

Face search strategy	Number of processed windows
Classical FSS [5]	5792
Improved FSS	295
Improved FSS and 2 Haar-like features	64
Improved FSS and 4 Haar-like features	13
Improved FSS and 6 Haar-like features	7

The improved FSS in conjunction with the application of Haar-like features allows accelerating FD process by diminishing of the scanning sub-images number especially when input images contain large faces.

4 Conclusions and Future Works

This paper presents some face candidate selection algorithms and improved neural network-based method. Face candidate detection is performed using the skin color and Haar-like features. The improved active training algorithm allows neural network working with the relatively small representative training set. The proposed face search strategy accelerates the face detection process using the inverse image scale pyramid, adaptive window scanning step, window acceptance, and is perfectly suitable for input images with large faces.

Our future research will be focused on further speedup of the face search process by construction of classifier's cascade, like in [3], where the final strong classifier (retinally connected neural network) is transformed into the cascade of modular neural networks. We're also transforming our Matlab routines into C++ application using OpenCV library [9].

Acknowledgements

The authors are grateful for the support to the Fundamental Researches State Fund of Ukraine, as the above results were obtained as a part of the research project "Development of methods and algorithms of face detection and recognition for real-time video-supervision systems".

References

1. Ming Hsuan Yang: Recent Advances in Face Detection, IEEE ICPR 2004 Tutorial, Cambridge, United Kingdom (2004)
2. V. Vezhnevets, V. Sazonov, A. Andreeva: A Survey on Pixel-Based Skin Color Detection Techniques, Graphics and Media Laboratory, Faculty of Computational Mathematics and Cybernetics, Moscow State University, Moscow, Russia (2003)
3. P. Viola, M. Jones: Robust Real-Time Face Detection, International Journal of Computer Vision 57(2) (2004) 137–154
4. K. K. Sung and T. Poggio: Example-based learning for view-based human face detection, IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 20, No. 1 (1998) 39–51
5. H. Rowley, S. Baluja, and T. Kanade: Neural network-based face detection. In IEEE Patt. Anal. Mach. Intell., Volume 20 (1998) 22–38
6. Peter Peer, Jure Kovac, Franc Solina: Human Skin Colour Clustering for Face Detection, EUROCON 2003 - International Conference on Computer as a Tool, Eds. B. Zajc, Volume 2. Ljubljana, Slovenia (2003) 144–148
7. CBCL Face Database #1. MIT Center For Biological and Computation Learning. <http://www.ai.mit.edu/projects/cbcl>
8. Raphael Feraud, Olivier J. Bernier, Jean-Emmanuel Viallet, and Michel Collobert: A Fast and Accurate Face Detector Based on Neural Networks, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No. 1 (2001)
9. <http://sourceforge.net/projects/opencv/>

Artificial Neural Networks based Approaches: Simulation Toys or Real Solutions?

Kurosh Madani

Images, Signals, and Intelligent System Laboratory (LISSI / EA 3956)
Paris-XII University, Senart Institute of Technology
Avenue Pierre Point, Lieusaint, 77127, France
madani@univ-paris12.fr

Abstract. If over the past decades understanding, modeling and improvement of learning and generalization capabilities of Artificial Neural networks have been subject to a particular attention, nowadays, these connectionist models have to face up to new challenges dealing with industrial requirements and real-world dilemmas. The target is to explore this pertinent point through a set of ANN based solutions developed in order to defy a number of applicative challenges dealing with industrial and real world requirements.

1 Introduction

Overcoming limitations of conventional approaches thank to their learning and generalization capabilities, Artificial Neural Networks (ANN) made appear a number of expectations to design “intelligent” information processing systems. If over the past decades understanding, modeling and improvement of learning and generalization capabilities of these bio-inspired models have been subject to a particular attention, nowadays, these connectionist models have to face up to new challenges dealing with industrial requirements and real-world dilemmas. In other words, now the time is to fulfill the following question: do ANN based approaches remain simulation toys or they reveal real solutions for reaching farther technological borders?

The present paper aims to contribute in discussion around the possible answer to the above-formulated question. The target is to discuss and explore this pertinent question through a set of ANN based solutions developed in order to defy a number of applicative challenges dealing with industrial and real world requirements.

The present paper is organized in following way: the next section will introduce the first application of ANN based approaches dealing with “automated visual inspection” in industrial production of VLSI devices. Section 3, will present another industrial application of ANN based concepts focusing fault detection and defects’ classification in high-tech optical devices production. Finally the last section of this paper will conclude the present article and discuss a number of perspectives.

2 ANN based Probe Mark Inspection in VLSI Chips Production

One of the main steps in VLSI circuit production is the testing step. This step verifies if the final product (VLSI circuit) operates correctly or not. The verification is performed thanks to a set of characteristic input signals (stimulus) and associated responses obtained from the circuit under test. A set of such stimulus signals and associated circuit's responses are called test vectors. Test vectors are delivered to the circuit and the circuit's responses to those inputs are catch through standard or test dedicated Input-Output pads (I/O pads) called also vias. As in the testing step, the circuit is not yet packaged, the test task is performed by units, which are called probers including a set of probes performing the communication with the circuit. The problem is related to the fact that the probes of the prober may damage the circuit under test. So, an additional step consists of inspecting the circuit's area to verify vias (I/O pads) status after circuit's testing: this operation is called developed Probe Mark Inspection (PMI). Fig.1-a shows a picture of probes relative to such probers. Fig.1-b gives examples of faulty and correct vias.

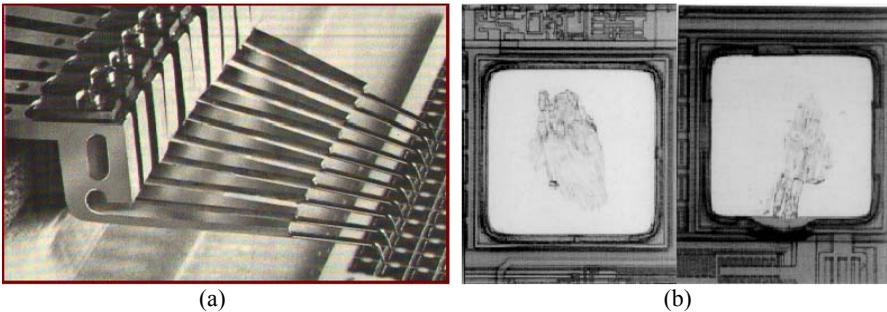


Fig. 1. Photograph giving an example of probes in industrial prober (a). Example of probe impact: correct and faulty (b).

Many prober constructors had already developed PMI software based on conventional pattern recognition algorithms with little success]. The difficulty is related to the compromise between real time execution (production constraints) and methods reliability. In fact, even sophisticated hardware using DSPs and ASICs specialized in image processing are not able to perform sufficiently well to convince industrials to switch from human operator (expert) defects recognition to electronically automatic PMI. That's why a neural network based solution has been developed and implemented on ZISC-036 neuro-processor, for the IBM Essonnes plant. The main advantages of developed solutions are real-time control and high reliability in fault detection and classification tasks. Our automatic intelligent PMI application, detailed in [1] and [2], consists of software and a PC equipped with this neural board, a video acquisition board connected to a camera and a GPIB control board connected to a wafer prober system. Its goal is image analysis and prober control.

The IBM ZISC-036 (see [3] and [4]) is a parallel neural processor based on the RCE and KNN algorithms. Each chip is capable of performing up to 250 000 recognitions per second. Thanks to the integration of an incremental learning

algorithm, this circuit is very easy to program in order to develop applications; a very few number of functions (about ten functions) are necessary to control it. Each ZISC-036 like neuron implements two kinds of distance metrics called L1 and LSUP respectively. Relations (1) and (2) define the above-mentioned distance metrics where P_i represents the memorized prototype and V_i is the input pattern. The first one (L1) corresponds to a polyhedral volume influence field and the second (LSUP) to a hyper-cubical influence field.

$$L1 : dist = \sum_{i=0}^n |V_i - P_i| \quad (1)$$

$$LSUP : dist = \max_{i=0..n} |V_i - P_i| \quad (2)$$

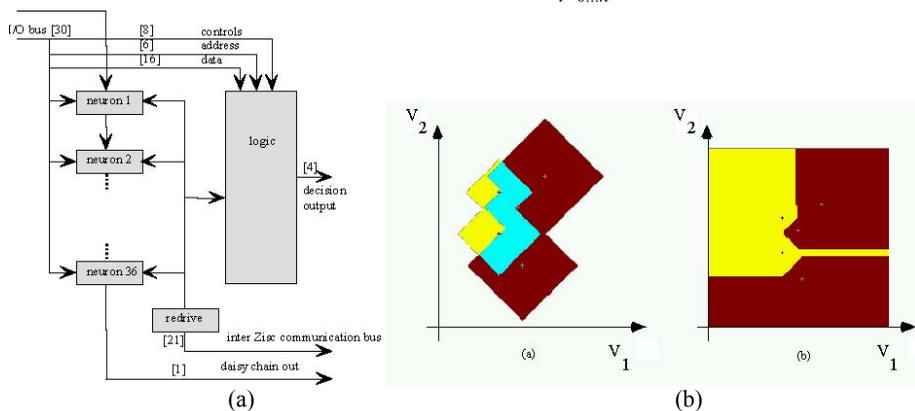


Fig. 2. IBM ZISC-036 chip's bloc diagram (a) and an example of input feature space mapping in a 2-D space using ROI and 1-NN modes, using norm L1 (b).

Figure 2 gives the ZISC-036 chip's bloc diagram and an example of input feature space mapping in a 2-D space. A 16 bit data bus handles input vectors as well as other data transfers (such as category and distance), and chip controls. Within the chip, controlled access to various data in the network is performed through a 6-bit address bus. ZISC-036 is composed of 36 neurons. This chip is fully cascadable which allows the use of as many neurons as the user needs (a PCI board is available with a 684 neurons). A neuron is an element, which is able to:

- memorize a prototype (64 components coded on 8 bits), the associated category (14 bits), an influence field (14 bits) and a context (7 bits),
- compute the distance, based on the selected norm (norm L1 given by relation or LSUP) between its memorized prototype and the input vector (the distance is coded on fourteen bits),
- compare the computed distance with the influence fields,
- communicate with other neurons (in order to find the minimum distance, category, etc.),
- adjust its influence field (during learning phase).

Two kinds of registers hold information in ZISC-036 architecture: global registers and neuron registers. Global registers hold information for the device or for the full network (when several devices are cascaded). There are four global registers

implemented in ZISC-036: a 16-bits Control & Status Register (CSR), a 8-bits Global Context Register (GCR), a 14-bits Min. Influence Field register (MIF) and a 14-bits Max. Influence Field register (MAF). Neuron registers hold local data for each neuron. Each neuron includes five neuron registers: Neuron Weight Register (NWR), which is a 64-by-8 bytes register, a 8-bits Neuron Context Register (NCR), Category register (CAT), Distance register (DIST) and Neuron Actual Influence Field register (NAIF). The last three registers are both 14-bits registers. Association of a context to neurons is an interesting concept, which allows the network to be divided in several subsets of neurons. Global Context Register (GCR) and Neuron Context Register (NCR) hold information relative to such subdivision at network and neuron levels respectively. Up to 127 contexts can be defined.

The process of analyzing a probe mark can be described as following: the PC controls the prober to move the chuck so that the via to inspect is precisely located under the camera; an image of the via is taken through the video acquisition board, then, the ZISC-036 based PMI:

- finds the via on the image,
- checks the integrity of the border (for damage) of via,
- locates the impact in the via and estimates its surface for statistics.

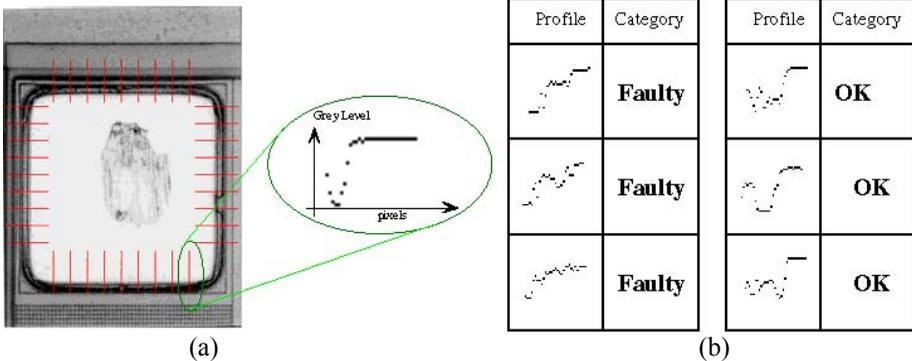


Fig. 3. Example of profiles extraction after via centring process (a). Example of profiles to category association during the learning phase (b).

All vias of a tested wafer are inspected and analyzed. At the end of the process, the system shows a wafer map which presents the results and statistics on the probe quality and its alignment with the wafer. All the defects are memorized in a log file. In summary, the detection and classification tasks of our PMI application are done in three steps: via localization in the acquired image, mark size estimation and probe impact classification (good, bad or none).

The method, which was retained, is based on profiles analysis using kernel functions based ANN. Each extracted profile of the image (using a square shape, figures 3 and 4) is compared to a reference learned database in which each profile is associated with its appropriated category. Different categories, related to different needed features (as: size, functional signature, etc).

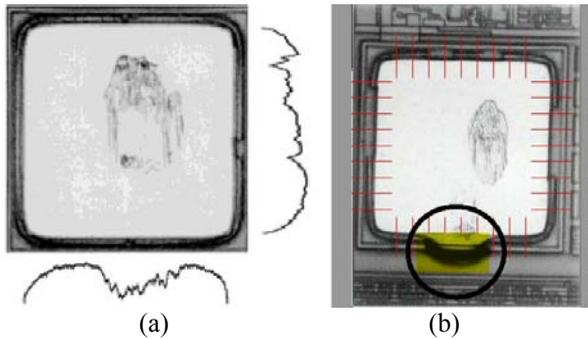


Fig. 4. Profiles extraction for size and localization of the probe mark (a). Experimental result showing a fault detection and its localization in the via (b).

Experiments on different kinds of chips and on various probe defects have proven the efficiency of the neural approach to this kind of perception problem. The developed intelligent PMI system outperformed the best solutions offered by competitors by 30%: the best response time per via obtained using other wafer probers was about 600 ms and our neural based system analyzes one via every 400 ms, 300 of which were taken for the mechanical movements. Measures showed that the defect recognition neural module's execution time was negligible compared to the time spent for mechanical movements, as well as for the image acquisition (a ratio of 12 to 1 on any via). This application is presently inserted on a high throughput production line.

3 Automated Classification of High-Tech Optical Devices' Defects in Industrial Production

Fault diagnosis in industrial environment is a challenging but crucial task, since it ensures products' nominal specification and manufacturing control. Concerning High-Tech optical industry, a major step for high-quality optical devices' faults diagnosis concerns scratches and digs defects detection and characterization in such products. These kinds of aesthetic flaws, shaped during different manufacturing steps, could provoke harmful effects on optical devices' functional specificities, as well as on their optical performances by generating undesirable scatter light, which could seriously damage the expected optical features. A reliable diagnosis of these defects becomes therefore a crucial task to ensure products' nominal specification. Moreover, such diagnosis is strongly motivated by manufacturing process correction requirements in order to guarantee mass production quality with the aim of maintaining acceptable production yield.

Unfortunately, detecting and measuring such defects is still a challenging problem in production conditions and the few available automatic control solutions remain ineffective. That's why, in most of cases, the diagnosis is performed on the basis of a human expert based visual inspection of the whole production. However, this conventionally used solution suffers from several acute restrictions related to human

operator's intrinsic limitations (reduced sensitivity for very small defects, detection exhaustiveness alteration due to attentiveness shrinkage, operator's tiredness and weariness due to repetitive nature of fault detection and fault diagnosis tasks). Figure 5 gives an example of High-Tech optical products, showing four optical filters. The same figure shows an example of visual inspection process of the aforementioned defects, requiring expert knowledge and a consequent delay [5].



Fig. 5. Example of High-Tech optical devices performing optical filtering (a) and the visual fault detection, performed by an expert (b).

To construct an automatic diagnosis system, we propose an approach based on three main operations: detection, classification and decision. Our motivation to adopt the approach dissociating detection and diagnosis tasks is based on requirement relative to the frame of industrial production. In fact, two complementary options could be required in industrial production environment. The first is inherent to mass production where it is not always necessary to diagnose whole manufactured products during the production, but it is crucial to detect the presence of defects in order to state if the number of defects is conform to the process' intrinsic limitations. However, at the same time, diagnosis ability could help to state (offline) if detected defects are due to intrinsic limitations of the used manufacturing process or a number of them correspond to different derivations. The second situation is specific to High-Tech products manufacturing requirements, where additionally to systematic defect detection it is crucial to state on nature of detected defects in order to reach high-quality specifications.

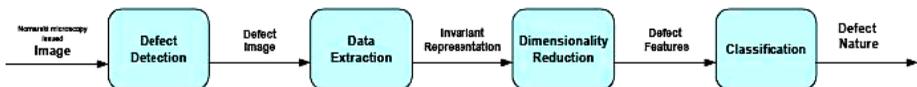


Fig. 6. Block diagram of the proposed optical devices diagnosis system.

To perform this challenging task, we choose to use neural network based techniques, which have shown many attractive features in complex patterns recognition and classifications. The outline of the process, we propose to use in order to carry out the defect classification is shown in the diagram of figure 6. As one could remark, the whole system includes four main stages (tasks): defect detection stage, data extraction module, dimensionality reduction stage and classification module [5]. The detection approach is based on Nomarski's microscopy issued imaging (NMI) [6], [7], [8], [9]. This method provides robust detection and reliable measurement of

outward defects (essentially scratches and digs defects), making plausible a fully automatic inspection of optical products.

The aim of the second stage is to extract defects images from Nomarski detector issued digital image. A new method has been proposed including four phases:

- Pre-processing: Nomarski issued digital image transformation in order to reduce lighting heterogeneity influence and to enhance the aimed defects' visibility,
- Adaptive matching: adaptive process to match defects,
- Filtering and segmentation: noise removal and defects' outlines characterization.
- Defect image extraction: correct defect representation construction.

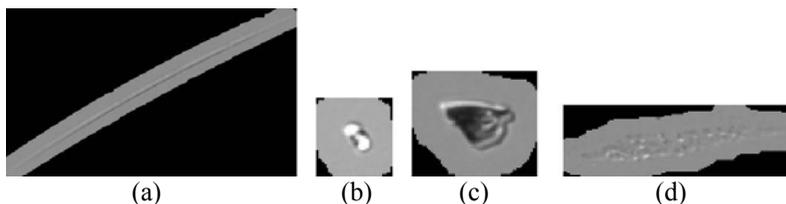


Fig. 7. Examples of images of different characteristic items obtained after “data extraction” stage: (a) scratch; (b) dig; (c) dust; (d) cleaning marks.

The dimensionality reduction task consists on constructing a set of lower dimension homogenous invariant (regarding translation and rotation) representation of characteristic items obtained from “data extraction” stage. In fact, processing high-dimensional data may induce several problems among which, the exponentially increase of the number of samples, required to reach a predefined level of precision in approximation tasks, with the dimension of considered space [10]. Concerning a ANN based classification, taking into account the above-mentioned problem, one can intuitively understand that the number of samples needed to properly learning high-dimensional data becomes quickly too large to be collected by a real systems.

A first data dimensionality reduction is performed using “Fourier-Mellin” transformation as it provides invariant descriptors, which are considered to have good coding capacity in classification tasks [11], [12], [13] and [14]. In fact, because of different sizes of items' images and their relative positions (due to translation and rotation) it is necessary to have a “normalize” representation for classification stage's input patterns. After “Fourier-Mellin” transformation each item image obtained from the second stage is represented as a 13-D vector. Then, such 13-D translation-rotation invariant vector is normalized thanks to a centring-reducing transformation, modifying each feature F_i conformably to the relation (3), where M is the mean value of the feature F_i over the database and σ its standard deviation.

$$F_i = \frac{F_i - M}{\sigma} \quad (3)$$

A second data dimensionality reduction is then performed using a projection based approach from a high-dimensional space to a lower-dimensional space. Several

techniques have been studied, among which, Kohonen Self-Organizing Map (SOM), Curvilinear Component Analysis (CCA) and Curvilinear Distance Analysis (CDA).

These aspects are presented and discussed in [15]. SOM is often used in industrial engineering [16], [17] to characterize high-dimensional data or to carry out classification tasks. Its main advantage is to offer an additional “pre-classification” ability. However, it suffers from a number of drawbacks: first the SOM’s topology is static and should be fixed a priori; moreover the method defines only a discrete nonlinear subspace; finally algorithm is computationally too expensive to be practically applied for projection space dimension higher than 3 [15], [18]. Concerning CCA [19], if it is able to reproduce the topology of an n-dimension original space in a new p-dimension space (where $p < n$) without any a priori configuration of the topology (constituting its advantage regarding SOM), it remains essentially a linear projection (its main drawback regarding complex classification tasks) [20]. CDA involves curvilinear distances allowing dealing with non-linear manifolds (its main advantage) [21]. Finally, the classification stage, including an unsupervised learning based “pre-classification” stage and a supervised learning based classifier, performs the defects’ classification operation.

Table 1. Validation results relative to the MLP classification performances.

Training database dimensionality	Correct classification	Standard Deviation
13	76 %	1.33 %
2	94 %	0.87 %

The validation of proposed approach has been done using data issued from real industrial production process [15]. Two Multi-Layer Perceptron (MLP) based classifiers, one classifying 13-D data and the other classifying 2-D data have been implemented. The first MLP structure includes 13 input neurons, 35 neurons in hidden layer, and 2 output neurons (13-35-2 MLP) and the second one engages 2 input neurons, 35 neurons in hidden layer, and 2 output neurons (2-25-2 MLP). For both of above-indicated structures, the training was achieved 20. Results are presented in Table 4. These results clearly prove that the considered classification problem is simplified, when properly reformulated in an appropriated lower dimensional space.

4 Conclusion and Perspectives

The main goal of this paper was to show how ANN models could be sources of inspiration in emergence of real industrial solutions. The presented applications and issued results show the significant potentiality of connectionist architectures for designing real world applications dealing with complex industrial dilemmas.

Acknowledgements

I would like to express my gratitude to Dr. Veronique Amarger, Dr. Abdennasser Chebira, Dr. Amine Chohra and Dr. Christophe Sabourin, working as staff members of my research team (and members of LISSI Lab.) involved in presented works. Also, I would like to express my gratitude as well to Dr. Ghislain De Tremiolles, who have worked as Ph.D. student on ZISC-036 neuro-processor Based Probe Mark Inspection as to IBM-France that supported this project. In the same way, I would like express gratitude to Mr. Mathieu Voiry my Ph.D. student working with SAGEM on Automated Classification of High-Tech Optical Devices' Defects for his valuable work allowing accomplished progresses in this project. Finally, I would acknowledge SAGEM industrial group for supporting Mr. Voiry and this challenging project.

References

1. G. Trémiolles (de), P. Tannhof, B. Plougonven, C. Demarigny, K. Madani, Visual Probe Mark Inspection, using Hardware Implementation of Artificial Neural Networks, in VLSI Production, Lecture Notes in Computer Science - Biological and Artificial Computation : From Neuroscience to Technology, Edited by : Jose Mira, Roberto M. Diaz and Joan Cabestany - Springer Verlag Berlin Heidelberg, pp 1374-1383, 1997, (1997)
2. K. Madani, G. De Tremiolles, P. Tanhoff, Image processing using RBF like neural networks : A ZISC-036 based fully parallel implementation solving real world and real complexity industrial problems, Journal of Applied Intelligence N°18, 2003, Kluwer Academic Publishers, pp. 195-231, (2004)
3. G. Trémiolles (de), K. Madani, P. Tannhof, A New Approach to Radial Basis Function's like Artificial Neural Networks , NeuroFuzzy'96, IEEE European Workshop, Vol. 6 N° 2, pp 735-745, April 16 to 18, Prague, Czech Republic, (1996)
4. G. De Tremiolles, Contribution à l'étude théorique des modèles neuromimétiques et à leur validation expérimentale: mise en œuvre d'applications industrielles ("Contribution to the theoretical study of neuromimetic models and to their experimental validation: use in industrial applications"), Ph.D. Report, University of PARIS XII – Val de Marne, (1998)
5. K. Madani, M. Voiry, V. Amarger, N. Kanaoui, A. Chohra, F. Houbre "Computer Aided Diagnosis Using Soft-Computing Techniques and Image's Issued Representation : Application to Biomedical and Industrial Fields", COMPUTING, Special Issue on Intelligent Computing, Vol.5, Issue 3, ISSN 1727-6209, pp. 43-53, (2007)
6. P. Bouchareine, Métrologie des Surfaces. Techniques de l'Ingénieur, vol. R1390, 1999 (in French).
7. S. Chatterjee, Design Considerations and Fabrication Techniques of Nomarski Reflection Microscope. Optical Engineering, vol. 42, no. 8, pp. 2202-2212, (2003).
8. M. Voiry, F. Houbre, V. Amarger, and K. Madani, Toward Surface Imperfections Diagnosis Using Optical Microscopy Imaging in Industrial Environment, Workshop on Advanced Control and Diagnosis, Mulhouse, France, pp. 139-144, (2005).
9. M. Voiry, K. Madani, V. Amarger, F. Houbre, Toward Automatic Defects' Clustering in Industrial Production Process Combining Optical detection and Unsupervised Artificial Neural Network techniques, Artificial Neural Networks and Intelligent Information Processing, INSTICC Press, N°ISBN: 978-972-8865-68-9, pp. 25-34, (2006).
10. M. Verleysen: Learning high-dimensional data. LFTNC'2001 - NATO Advanced Research Workshop on Limitations and Future Trends in Neural Computing (2001).

11. A. Choksuriwong, H. Laurent, and B. Emile, Comparison of invariant descriptors for object recognition. IEEE International Conference on Image Processing (ICIP) pp. 377-380, (2005)
12. S. Derrode, Représentation de Formes Planes à Niveaux de Gris par Différentes Approximations de Fourier-Mellin Analytique en vue d'Indexation de Bases d'Images. Phd Thesis Rennes I Univ., (in French) (2005)
13. F. Ghorbel, A Complete Invariant Description for Gray Level Images by the Harmonic Analysis Approach. Pattern Recognition, vol. 15, pp. 1043-1051, (1994)
14. G. Ravichandran and M. Trivedi, Circular-Mellin features for texture segmentation. IEEE Trans. Image Processing, vol. 4, pp. 1629-1640, (1995)
15. M. Voiry, K. Madani, V. Amarger, J. Bernier, Impact of Data Dimensionality Reduction on Neural Based Classification: Application to Industrial Defects, ANNIIP 2007 (to be published in May, 2007).
16. T. Kohonen, E. Oja, O. Simula, A. Visa, and J. Kangas, Engineering Applications of the Self-Organizing Maps. Proceedings of the IEEE, vol. 84, no. 10, pp. 1358-1384, (1996)
17. J. Heikkonen and J. Lampinen: Building Industrial Applications with Neural Networks., Proc. European Symposium on Intelligent Techniques, ESIT'99 (1999).
18. T. Kohonen, Self Organizing Maps, 3rd edition, Berlin: Springer, (2001)
19. P. Demartines and J. Héroult: Vector Quantization and Projection Neural Network, Lecture Notes in Computer Science, vol. 686, International Workshop on Artificial Neural Networks, p.328-333 (1993).
20. P. Demartines and J. Héroult: CCA : "Curvilinear Component Analysis", Proceedings of 15th workshop GRETSI (1995).
21. J. A. Lee, A. Lendasse, N. Donckers, and M. Verleysen: A Robust Nonlinear Projection Method, European Symposium on Artificial Neural Networks - ESANN'2000 (2000).

Author Index

Amar, C.	30	Sveda, M.	76
Amarger, V.	56	Tan, J.	99
Bellil, W.	30	Vázquez, R.	3
Bernier, J.	56	Voiry, M.	56
Budnyk, I.	38	Volná, E.	13
Chebira, A.	38	Zhang, G.	66
Chohra, A.	107		
Clemente, R.	83		
Dorogush, A.	91		
Farrell, R.	66		
Górriz, J.	83		
Hung, P.	66		
Inestroaa, J.	83		
Ishikawa, S.	99		
Islam, M.	48		
Keck, I.	83		
Kim, H.	99		
Kurylyak, Y.	107		
Madani, K.	38, 56, 107, 115		
McLoone, S.	66		
Mukhin, V.	21		
Nakano, S.	99		
Nassabay, S.	83		
Othmani, M.	30		
Otsuka, Y.	99		
Paliy, I.	107		
Podenok, L.	91		
Puntonet, C.	83		
Rysavy, O.	76		
Sachenko, A.	107		
Sadykhov, R.	91		
Sánchez, M.	66		
Shimizu, H.	99		
Shinomiya, T.	99		
Shyrochin, V.	21		
Sossa, H.	3		



Proceedings of the
3rd International Workshop on
Artificial Neural Networks and Intelligent Information
Processing - ANNIIP 2007
ISBN: 978-972-8865-86-3
<http://www.icinco.org>