Using Testor Theory to Reduce the Dimension of Neural Network Models

Roberto A. Vázquez [1] and Salvador Godoy-Calderón [2]

Computing Research Center CIC-IPN,
Av. Juan de Dios Bátiz, esquina con Miguel Othón de Mendizábal,
Mexico City, 07738, MEXICO.
Contact: [1]ravem@ipn.mx , [2]sgodoyc@cic.ipn.mx

(Paper received on July 10, 2007, accepted on September 1, 2007)

Abstract. Most of the Neural Network models proposed during the last few years are capable of solving several complex problems such as recognition, forecast or reconstruction of different phenomena. A crucial feature of these models is that they tackle a wide variety of classification problems, and although these models work accurately within a limited particular context, they require numerous resources; this makes its efficient hardware implementation nearly impossible. In this paper we propose a novel alternative which uses Testor Theory, which is a useful tool in pattern recognition within the logical-combinatorial approach, to reduce the dimension of neural network models. We test the accuracy of the proposed method by recognizing real-world objects in images. We show that, under some circumstances, it is possible to reduce the dimension of a neural network without affecting its capability to solve classification problems or modifying its effectiveness.

1 Introduction

Humans have several complex cognitive capabilities such as memorizing, recalling, learning and recognizing. In the last 50 years, scientists and researchers of different communities have strived to implement these capabilities into a computer. Along these years, several approaches for achieving that goal have emerged, for example neural networks. Since the rebirth of neural networks, several models inspired in neurobiological processes have been proposed. Such models are often dedicated and incorporate some existing clustering or classification algorithm. Among these models, perhaps the most popular one is the feed-forward multi-layer perceptron trained with the back-propagation algorithm [10].

Other very popular neural models are associative memories. Several of these associative models have been proposed, for example: Anderson [6] presents a simple neural network generating an interactive memory; Kohonen [7] presents an unsupervised learning network as an explanation of the existence of ordered maps in the brain. Other associative models can be found in [3], [9], [26], [27], [28], [29], [32] and [33]. Advantages of neural networks are: adaptability, robustness, and ease of implementation in software.

Most of the neural network models are capable of solving several kinds of complex problems such as face recognition, object recognition, plate recognition, hand-writing recognition and some other classification problems. Despite the fact that these models work accurately, they require too many resources, thus making their efficient implementation on hardware nearly impossible.

In this paper we propose a novel alternative to reduce the dimension of neural network models by using Testor Theory, which is a useful tool in pattern recognition under the logical-combinatorial approach. We test the accuracy of the proposal by solving an object recognition problem using several images of realistic objects. In the experiments, we expect to reduce the dimension of the neural network models used for solving the problem by reducing the dimensionality of the features they take as input.

2 A Survey of Dimensional Reduction Techniques

When working with high-dimensional datasets it is often the case that not all the measured variables have the same "relevance" for understanding the underlying phenomena of interest. Certain computationally-expensive methods can construct predictive models with high accuracy from high-dimensional data. Still reducing the dimension of the original data prior to any modeling is of the outmost interest. This reduction process directly modifies the dimension of any neural network or associative memory.

In mathematical terms, the problem under study can be stated as follows: given a p-dimensional random variable $\mathbf{x}=(x_1,\ldots,x_p)^T$, find a lower dimensional representation of it, $\mathbf{s}=(s_1,\ldots,s_k)^T$ with $k<p$, that captures the content in the original data, according to some criteria, usually stated within a supervision/learning/training set.

Several statistical techniques, such as principal component analysis [11], [13] and factor analysis have been proposed for achieving this dimensional reduction. Although these techniques based on second-order statistics are computationally expensive they are widely used. In other cases, when dealing with statistically normal variables (those with mean $=0$) the covariance matrix already contains all the necessary information about the data. Second-order methods for dimensional reduction are relatively easy to code, as they require only simple matrix operations. However, many datasets of interest are not suitable for studying within a Gaussian distribution. For those cases, higher-order dimensional reduction methods, using information not contained in the covariance matrix, are more appropriate. Examples of these methods are independent component analysis and projection pursuit [20]. Another interesting method is random projections, which is a simple, yet powerful dimensional reduction technique that uses random matrices to project data into lower dimensional spaces [12], [15], [17], [21].

Some very useful non-statistical methods for dimensional reduction were proposed in the mid-fifties in the former Soviet Union and were later developed in other Eastern European countries and in Cuba. These logical-combinatorial methods are based on Testor Theory (formerly referred to as "Test Theory") and use the concepts of Testor and Non-reducible Testor [24] which were introduced for the first time by Yablonskii and Cheguis [1], [2] and later applied to classification problems by Dimitriev et al [5]. Testor Theory methods are used in this research to reduce the dimensionality of the data taken as input to different neural network models.

3 Dimensional Reduction Using Testor Theory

In this section, some aspects of Testor Theory applied to feature selection and dimensional reduction problems are applied. We also describe a technique, devised by Godoy-Calderón et al [25], used to compute a set of special kind of Testors called Super-Testors used to reduce the dimension of a neural network model.

3.1 Basics of Testor Theory

This section is based on [30]. In the framework of the logical-combinatorial pattern recognition [16], [19], [22], feature selection or dimensional reduction could be made by using Non-reducible Testors [23]. If $\Re$ is the whole set of attributes of the objects under study and thus the corresponding patterns are $\Re$-dimensional, a Testor is defined as follows:

Definition 1. A feature subset $\tau \subseteq \Re$ is a Testor if and only if when all features, except those from τ, are eliminated from the descriptions, no pair of similar subdescriptions remain in different classes. This definition indicates that a Testor is a feature subset, which allows complete differentiation of objects from different classes. Within the set of all Testors, there are some which are Non-reducible. These kind of Testors are called Typical Testors and are defined as follows:

Definition 2. A feature subset $\tau \subseteq \Re$ is a Typical Testor if and only if τ is a Testor and there is no other Testor τ' such that $\tau' \subset \tau$. This definition indicates that a Typical Testor is a Testor where every feature is essential, this is, if any of them are eliminated the resultant set is not a Testor.

The dimensional reduction approach based on Testor Theory was first proposed by Dimitriev [5] and the basic idea is the following: A Testor is a feature subset, which does not induce confusion between any pair of sub-descriptions of objects from different classes. Moving, from a Testor to a Typical Testor (eliminating features, when it is possible) we get an irreducible combination of features, where each feature is essential in order to keep differences between classes.

3.2 Testors and fuzzy classification of objects.

Let O be a set of objects denoted by o_i, each object is described in terms of a set of n features denoted by $\mathbf{x}^i = \delta(o_i)$ and these objects are grouped into c classes. Let $M = [m_{ik}]_{p \times c}$ be the membership matrix where p is the number of descriptions, c the number of classes and $m_{ik} = \mu_k(\delta(o_i))$ the membership of object o_i to class k given by:

$$\mu_k(o_i) = \bigvee_{j=1}^{p} \left(\beta\left(\delta(o_i), \delta(o_j)\right) * \mu_k\left(\delta(o_j)\right) \right) \tag{1}$$

where $\beta\left(\delta(o_i), \delta(o_j)\right)$ is any similarity function between o_i and o_j which yields a result in the range $[0,1]$. Confusion between objects in different classes depends on their membership to each class. An object o_i is confused between two classes a and b if and only if $\delta(o_i) \in (a \cap b)$.

Definition 3. The Discrimination Error ε of an object o_i is given by:

$$\varepsilon\left(\delta(o_i)\right) = \sum_{i,j \in [1,c]} \mu_{a_i \cap b_j}\left(\delta(o_i)\right) \tag{2}$$

this is the sum of its membership to any intersection between classes. The Cumulative Discrimination Error $\hat{\varepsilon}$ is given by:

$$\hat{\varepsilon} = \sum_{i=0}^{p} \varepsilon\left(\delta(o_i)\right) \tag{3}$$

Definition 4. A feature subset $\tau \subseteq \Re$ is a Testor with Level n if and only if $\hat{\varepsilon} = \sum_{i=0}^{p} \varepsilon\left(\delta(o_i)\big|_{\tau}\right) = n$. When $n = 0$ this definition is equivalent to Definition 1.

Definition 4 allows any subset of $\Re$ to be a Testor but with a different level. Of course, the most interesting Testors are those which have level zero.

4 Dimensional Reduction of a Neural Network Model

Now we will show how a neural model, which solves a supervised classification problem, can be optimized by using Super-Testors. In this paper several neural network models used for solving object recognition problems. as in [31], were optimized.

4.1 Associative Memories

Let $\left(x^{\xi}, i\right)_{\xi=1}^{p}, x^{\xi} \in \Re^{n}, i = 1,\ldots,c$ be a set of p-fundamental couples (SFC) formed by a pattern x^{ξ} and its corresponding class-index i. We want to build an associative memory **M**, using this SFC, that allows us to classify the patterns into their corresponding classes, i.e. $\mathrm{M} \otimes x^{\xi} = i$ for ξ=1,..., p and which, even in the presence of distortions, classifies them adequately, i.e. $\mathrm{M} \otimes \tilde{x}^{\xi} = i$, where $\tilde{x}^{\xi}$ is an altered version of x^{ξ}. Operator $\otimes$ is chosen such that, when pattern x^{ξ} is operated with matrix **M**, it produces the corresponding index class of pattern x^{ξ}.

The associative memory **M** is built in terms of a ϕ function as follows:

$$\mathrm{M} = \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_c \end{bmatrix} \tag{4}$$

where each ϕ_i represents the i-th row of a matrix **M** and this function is a codification of all patterns belonging to class i. In this case ϕ_i is defined as:

$$\phi_i^j = \frac{\gamma_i^j + \lambda_i^j}{2} \tag{5}$$

where

$$\gamma_i^j = \bigvee_{\xi=1}^{p} \left(x_i^{\xi,j} \right) \tag{6}$$

and

$$\lambda_i^j = \bigwedge_{\xi=1}^{p} \left(x_i^{\xi,j} \right) \tag{7}$$

i stands for the object's class and j goes from 0 to n. the size of the pattern. It can be seen that the idea is to build a hyper-box enclosing patterns that belong to class i, by means of max "$\vee$" and min "$\wedge$" set operators.

Once the associative memory is trained, pattern classification is done as follows: Given a new pattern $x^{\xi} \in \Re^n$ the index class i is given as

$$i = \mathrm{M} \otimes x^{\xi} = \arg_{l} \left[\bigwedge_{l=1}^{m} \bigvee_{j=1}^{n} \left| m_{lj} - x_j \right| \right] \tag{8}$$

Operators $\vee \equiv \max$ and $\wedge \equiv \min$ execute morphological operations on the difference of the absolute values of element m_{lj} of M and the components x_j of pattern x^{ξ} to be classified. $\bigvee_{j=1}^{n} \left| m_{lj} - x_j \right|$ is a metric formed with the maximum between row l of M and pattern x^{ξ}, thus it can be written as $d(x, m_l) \equiv \bigvee_{j=1}^{n} \left| m_{lj} - x_j \right|$, m_l row of

M. With this metric, pattern classification is the process of assigning pattern x^{ξ} to the class whose row index is the nearest.

4.2 Optimizing the Associative Memory

The SFC can be seen as the set of objects denoted by o_i with cardinality p; each object is described in terms of a set of n features denoted by $\mathbf{x}^i = \delta(o_i)$ and grouped into c classes. Optimization of the associative model is done by computing the cumulative error of each testor τ, as shown in the next algorithm:

```
1. Select a feature subset $\tau$.
2. Compute membership matrix as in section 3.2, using equation 1 with
   Testor $\tau$.
3. Compute accumulative error using equation 3.
4. Go to step 1 until the cumulative error of the whole feature subset
   $\tau$ has been computed.
5. Select feature subset $\tau$, where level $n$ of the Testor is the
   minimum.
6. Finally with this feature subset $\tau$, train the associative memory.
```

5 Experimental Results

In this section, the proposal is tested with the set of realistic objects shown in Figure 1. Objects were not directly recognized by their images but instead from their invariant descriptions. The associative memory M is built with these invariant descriptions. Twenty images of each object in different positions, translations and scaled changes were used to get the invariant descriptions.

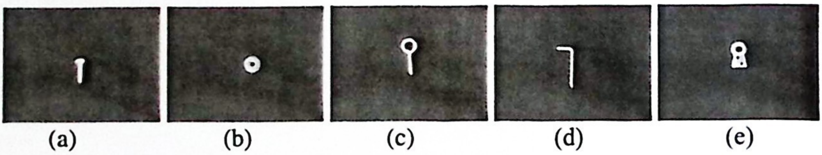
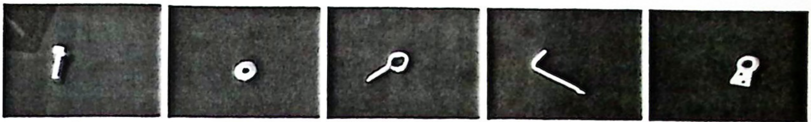


Fig. 1. The five objects used for training the associative memory. (a) A bolt, (b) A washer, (c) An eyebolt, (d) A hook, (e) A dovetail

A standard threshold [8] was applied to each of the 20 images in order to get their binary version. Small spurious regions from each image were eliminated by means of a standard-size filter [14]. Next, for each of the 20 images of each object (class) seven well-known Hu geometric moments invariant to translations, rotations and scale changes, were computed [4]. After applying the methodology described in Section 4.1, the associative memory M is:

$$\mathrm{M}=\begin{bmatrix} 0.4394 & 0.1598 & 0.0071 & 0.0028 & 1.96E-5 & 0.0011 & -8.47E-6 \\ 0.1900 & 8.72E-5 & 7.47E-6 & 1.28E-14 & 7.23E-14 & -2.93E-10 & -1.6E-14 \\ 0.7092 & 0.2895 & 0.1847 & 0.0730 & 0.0088 & 0.0394 & -0.0015 \\ 1.4309 & 1.6009 & 0.7944 & 0.2097 & 0.0831 & 0.1565 & 0.0118 \\ 0.2475 & 0.0190 & 2.5E-5 & 8.66E-5 & 4.82E-9 & 1.20E-5 & -1.4E-9 \end{bmatrix}$$

A new set of images was used to test the efficiency of the proposal. This set consisted of 100 images shown in Figure 2 (20 for each of the five objects), different from those used to get the associative memory M. Using this memory all objects of the set of images were put in their corresponding class. Thus, the performance of the proposal was 100%.



(a) (b) (c) (d) (e)

Fig. 2. The five objects used in the experiments: (a) A bolt. (b) A washer. (c) An eyebolt. (d) A hook. (e) A dovetail

In order to optimize the associative memory and reduce its dimensionality we applied the algorithm described in section 4.2. By applying this algorithm, we found that the feature subset τ, where level n of the testor is the minimum, was $\tau_{64}=\{x_1\}$, $\tau_{32}=\{x_2\}$ and $\tau_{96}=\{x_1,x_2\}$. After applying methodology described in Section 4.1, the corresponding associative memories are defined as:

$$\mathrm{M}_{64}=\begin{bmatrix} 0.4394 \\ 0.1900 \\ 0.7092 \\ 1.4309 \\ 0.2475 \end{bmatrix} \quad \mathrm{M}_{32}=\begin{bmatrix} 0.1598 \\ 8.72E-5 \\ 0.2896 \\ 1.6009 \\ 0.0190 \end{bmatrix} \quad \mathrm{M}_{94}=\begin{bmatrix} 0.4394 & 0.1598 \\ 0.1900 & 8.72E-5 \\ 0.7092 & 0.2896 \\ 1.4309 & 1.6009 \\ 0.2475 & 0.0190 \end{bmatrix}$$

Using associative memories M_{64} and M_{94} all objects of the set of images were put in their corresponding class. Thus, performance of the two optimized associative memories was of 100%. For the case of the associative memory M_{32} the performance was reduced to 84 %,.This result indicated that second moment of Hu is less representative than first moment . Table 1 summarizes the classification results for all associative memories tested.

Table 1. Comparative classification percentages of the associative memory with respect to the optimized associative memories

	M	M_{32}	M_{64}	M_{96}
Bolt	100%	90%	100%	100%
Washer	100%	100%	100%	100%
Eyebolt	100%	65%	100%	100%
Hook	100%	100%	100%	100%
Dovetail	100%	85%	100%	100%

In order to demonstrate that this technique is independent of the network architecture used, we performed several experiments using other neural networks. The same experiments were performed using the associative memories described in [32] and [27]. These associative memories use three different operators: **prom** operator, **med** operator and **median** operator. On the other hand we used the well-known multilayer neural network trained with the back-propagation algorithm as described in [10].

By applying the algorithm described in section 4.2, we found that the feature subset τ, where level n of the Testor is the least, was $\tau_{64} = \{x_1\}$, $\tau_{32} = \{x_2\}$ and $\tau_{96} = \{x_1, x_2\}$ for the different neural models.

For the case of **prom** operator, by using associative memories M_{64} and M_{94} all objects of the set of images were put in their corresponding class. Thus, the performance of the two optimized associative memories was of 100%. For the case of the associative memory M_{32} the performance was reduced to 88 %. Table 2 summarizes the classification results for all associative memories tested trained with **prom** operator.

Table 2. Comparative classification percentages of the associative memory using **prom** operator with respect to the optimized associative memories

	M	M_{32}	M_{64}	M_{96}
Bolt	100%	90%	100%	100%
Washer	100%	100%	100%	100%
Eyebolt	100%	65%	100%	100%
Hook	100%	100%	100%	100%
Dovetail	100%	85%	100%	100%

For the **med** operator case, by using associative memories M_{32}, M_{64} and M_{94} the performance was 82 %, which is the same result obtained with the complete description (e.g. with M). Table 3 summarizes the classification results for all associative memories tested and trained with **med** operator.

Table 3. Comparative classification percentages of the associative memory using **med** operator with respect to the optimized associative memories

	M	M_{32}	M_{64}	M_{96}
Bolt	100%	100%	100%	100%
Washer	100%	100%	100%	100%
Eyebolt	70%	70%	70%	70%
Hook	50%	50%	50%	50%
Dovetail	90%	90%	90%	90%

For the case of the **median** operator, by using associative memories M_{32}, M_{64} and M_{94} the performance was 95 %, again the same result obtained without reducing the network. Table 4 summarizes the classification results for all associative memories tested trained with **median** operator.

Table 4. Comparative classification percentages of the associative memory using **median** operator with respect to the optimized associative memories

	M	M_{32}	M_{64}	M_{96}
Bolt	100%	100%	100%	100%
Washer	100%	100%	100%	100%
Eyebolt	90%	90%	90%	90%
Hook	50%	50%	50%	50%
Dovetail	85%	85%	85%	85%

For the case of the multilayer neural network trained with the back-propagation algorithm, the performance for M_{32}, M_{64} and M_{94} was 89 %, 100% and 100%, while for M it was 98%. Table 5 summarizes the classification results for all neural networks trained with the back-propagation algorithm.

Table 5. Comparative classification percentages of the neural network with respect to the optimized neural network.

	M (7-4-1)	M_{32} (1-4-1)	M_{64} (1-4-1)	M_{96} (2-4-1)
Bolt	100%	95%	100%	100%
Washer	95%	100%	100%	100%
Eyebolt	95%	70%	100%	100%
Hook	100%	95%	100%	100%
Dovetail	100%	85%	100%	100%

6 Conclusions

In this paper we have described a new technique to optimize the size of an associative memory in a object recognition problem. This technique uses Testor theory, widely used in the logical-combinatorial approach to pattern recognition.

We describe an algorithm which allows us to calculate the most relevant features in the patterns used to train neural network models as associative memories and multilayer neural networks.

Throughout several experiments we test the accuracy of the proposal by using images of real objects. In those experiments first we train the neural models and then we show that by applying the procedure described in section 4.2 we can reduce the size of the neural models. We also show that in some cases the accuracy of the models is increased; like in the multilayer neural network, and in other cases the accuracy is the same. We are presently working in a comparison between classical dimensional reduction techniques and Testor Theory techniques applied to the neural network optimization task and also testing this approach with more complex objects.

Acknowledgments. Authors are grateful for the economic support provided by SIP-IPN under grant 20071432.

References

1. I.A. Cheguis, S.V. Yablonskii (1955), About testors for electrical outlines. *Usp. Mat. Nauk*, 4(66): 182-184 (in Russian).
2. I.A. Cheguis, S.V. Yablonskii(1958), Logical methods for controlling electrical systems. *Trudy MIAN ime V.A. Steklova*, LI :270-360 (in Russian).
3. K. Steinbuch (1961). Die Lernmatrix. *Kybernetik*, 1(1):26-45.
4. M. K. Hu (1962). Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8:179-187 .
5. Dimitriev A. N., Zhuravlev Y.I. and Krendeliev F.P. (1966), "About mathematical principles of objects and phenomena classification", *Diskretni Analiz* 7 , (In Russian).
6. J. A. Anderson (1972). A simple neural network generating an interactive memory. *Mathematical Biosciences*, 14:197-220.
7. T. Kohonen (1972). Correlation matrix memories. *IEEE Trans. on Computers*, 21(4):353-359.
8. N. Otsu (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on SMC*, 9(1):62-66.
9. J. J. Hopfield (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79: 2554-2558, 1982.
10. D. Rumelhart and J. McClelland (1986). Parallel distributed processing group. MIT Press.
11. I..T. Jolliffe(1986). *Principal Component Analysis*. Springer-Verlag.
12. H. Ritter and T. Kohonen (1989). Self-organizing semantic maps. *Biologial Cybernetic*, 61:241-251.
13. J.E. Jackson (1991). *A User's Guide to Principal Components*. New York: John Wiley and Sons.
14. R. Jain et al. *Machine Vision* (1995). McGraw-Hill. pp. 47-48.
15. S. Kaski (1997). *Data exploration using self-organizing maps*. PhD thesis, Helsinki University of Technology, Finland.

16. Alba Cabrera E. (1997), "*New extensions of testor concept for different types of similarity functions*" Ph. D. Thesis, ICIMAF, Cuba. (In Spanish).
17. S. Kaski (1998). Dimensionality reduction by random mapping: fast similarity computation for clustering. *Proc. IEEE International Joint Conference on Neural Networks,* 1:413-418.
18. R. Kohavi and G. John (1998). The wrapper approach. In H. Liu and H. Motoda, editors, *Feature Extraction, Construction and Selection: A Data Mining Perspective.* Springer Verlag.
19. J. Ruiz Shulcloper, A. Guzmán Arenas, and J. F. Martínez Trinidad (1999), *Logical Combinatorial Approach to Pattern Recognition. Feature Selection and Supervised Classification*, IPN, Mexico(In Spanish).
20. A. Hyvarinen (1999). Survey on independent component analysis. *Neural Computing Surveys*, 2:94-128.
21. T. Kohonen et al (2000). Self organization of massive document collection. *IEEE Transactions on Neural Networks,* 11(3):574-585.
22. J. F. Martínez Trinidad, and A. Guzmán-Arenas (2001), "The logical combinatorial approach to pattern recognition an overview through selected works", *Pattern Recognition,* 34(4):741-751.
23. M. Lazo-Cortes, J. Ruiz-Shulcloper and E. Alba-Cabrera (2001), "An overview of the evolution of the concept of testor", *Pattern Recognition,* 34(4):753-762.
24. J. Ruiz Shulcloper, and M.A. Abidi (2002). Logical Combinatorial Pattern Recognition: A Review. Recent Research Developments in Pattern Recognition, Pub. Transword Research Networks, USA, 133-176
25. S. Godoy-Calderón, M. Lazo-Cortés, J.F. Martínez-Trinidad (2003). A non-classical view of Coverings and its implications in the formalization of Pattern Recognition problems. WSEAS Transactions on Mathematics, 2, 1-2, 60-66.
26. 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.
27. 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.
28. H. Sossa and R. Barron (2003). New associative model for pattern recall in the presence of mixed noise. *In Proceedings of the fifth IASTED International Conference on Signal and Image Processing*, SIP2003. Acta Press 399:485-490.
29. H. Sossa, R. Barrón, R. A. Vázquez (2004). Transforming Fundamental set of Patterns to a Canonical Form to Improve Pattern Recall. *In proceedings of Ninth Ibero-American Conference on Artificial Intelligence* (IBERAMIA2004), LNAI 3315:687-696.
30. José Á. Santos, Ariel Carrasco and José F. Martínez (2004). Feature Selection using Typical Testors applied to Estimation of Stellar Parameters. *Computación y Sistemas*, 8(1):15-23.
31. R. A. Vázquez, H. Sossa and R. Barrón (2005). Object classification based on associative memories and midpoint operator. *Advances in Artificial Intelligence Theory* RCS 16:131-140, CIC-IPN.
32. R. A. Vázquez, H. Sossa (2006). Image categorization using associative memories. *Advances in Artificial Intelligence*, LNAI: 367-380.
33. R. A. Vázquez, H. Sossa and B. A. Garro (2006). A new bi-directional associative memory. *Progress in Pattern recognition, image analysis and application*, LNCS:549-558.
34. H. Sossa, R. Barrón and R. A. Vázquez (2004). *Real valued pattern classification based on extended associative memories*. Proceedings of 5th Mexican International Conference on Computer Science (ENC 2004), 20-24. IEEE Computer Society, edited by R. Baez, J. Marroquin and E. Chavez, pp 213-219.