A COMPARATIVE ANALYSIS BETWEEN CONVENTIONAL APPROACHES AND CONNECTIONIST METHODS IN PATTERN RECOGNITION TASKS

Oswaldo Ludwig Júnior [1]; Leizer Schnitman [2]; J.A.M.Felippe de Souza[3]; Herman Lepikson [1]

[1]Universidade Federal da Bahia
Department of Electrical Engineering
Salvador-BA-Brazil, e-mail: oludwig@terra.com.br; herman@ufba.br

[2] Faculdade de Ciência e Tecnologia - AREA1
Department of Electrical Engineering
Salvador-BA-Brazil, e-mail: leizer@area1.br

[3] Universidade da Beira Interior
Department of Electrical Engineering
Covilhã - Portugal, e-mail: felippe@dem.ubi.pt

ABSTRACT

The purpose of this work is to compare statistical and connectionist techniques for patterns recognition. Connectionist approach is based on feedforward artificial neural network, self-organizing maps or hybrid algorithms which are also compared. An alternative preprocessing method to pattern recognition problems is also suggested.

Keywords: self-organizing maps, neural networks, pattern recognition, statistics, artificial intelligence.

1. INTRODUCTION

The pattern recognition (PR) has been attracted several research interest. There are countless types of patterns, such as visual patterns, temporal pattern, logical pattern, … In a wide interpretation, it is possible consider that PR is present in any intelligent activity.

There are several approaches to the problem of PR, in such a way that it is possible to highlight some used methods such as the statistical approach [18], fuzzy [20], connectionist [8] and knowledge-based PR [19].

In the last decades significant progresses were obtained in this research area. These progresses allow the RP applications in several engineering areas [12, 15, 11]. Examples of applications that request efficient and robust techniques of PR can be highlighted: Classification of radar signals [8]; Data mining [5]; Bio-informatics [10]; Optical Character Recognition (OCR) [11]; Visual inspection for industrial automation [12]; Documents classification [13]; Biometrics recognition, including faces, iris or fingerprints [14]; Speech recognition [15].

Due to the success in these applications, the Artificial Intelligence approach becomes one of the most important tools to the successful application of PR.

In this paper, section 2 treats the statistical approach to PR and presents the V-C dimension. Section 3 describe the ANN feedforward and its use in PR tasks. The application of Self-Organizing Maps in PR is treated in section 4. Section 5 presents methods to pre-processing data, while Section 6 suggest a hybrid approach to RP. Conclusions are, finally, exposed in Section 7.

2. STATISTICAL APPROACH TO PR

This paper considers that classical algorithms for PR are based on statistics approach, which subdivide the classification problem in two different tasks: the features extraction and the comparison of these features with perfect models (i.e. noise free and representative of their respective patterns). These tasks are usually accomplished by two modules, which are denominated as features extractor and classifier [18, 19, 20].

The extracted features are commonly composed by a set of numeric values that should be enough for the appropriate representation of the input data, with respect to the classification task in subject. The features vector represents this set of values. Thus, a point in a features space can represent an object.

The model or prototype $\boldsymbol{\mu}_i$, considered as representative of a class or pattern $\boldsymbol{i}$, is usually obtained by a set $\boldsymbol{\aleph}$, composed by examples of features vectors belonging to $\boldsymbol{i}$, through the estimate of the medium vector:

$$\mu_i = \frac{\sum_{n=1}^{N} x_i[n]}{N} \cong E(x_i) = \lim_{N\to\infty} \frac{\sum_{n=1}^{N} x_i[n]}{N}, \quad \boldsymbol{N} = |\boldsymbol{\aleph}| \qquad (2.1)$$

The equation 2.1 is adapted to the batch training. However, some applications require recursive algorithms. In this case, the training is sequential and the computation of the medium vector $\boldsymbol{\mu}_i$ is given for:

$$\mu_i[n+1] = \frac{n}{n+1}\mu_i[n] + \frac{1}{n+1}x_i[n+1] \qquad (2.2)$$

The matching of an input data $\boldsymbol{x}$ with a specific pattern $\boldsymbol{i}$ is based on the "distance" measures, among $\boldsymbol{\mu}_i$ and $\boldsymbol{x}$. It means that the distance among two points in the features space can also be considered as the difference among two features vector. To the PR operation, a decision rule is adopted. It is based on the smallest distance.

There are several usual forms to check the distance $\boldsymbol{r}$ among $x=[x_1, x_2,..., x_n]^T$ and $\mu=[\mu_1, \mu_2,..., \mu_n]^T$. One can mention:

Euclidean distance:

$$r = \|x - \mu\| = \sqrt{\sum_n (x_n - \mu_n)^2} \quad (2.3)$$

Manhattan distance:

$$r = \sum_n |x_n - \mu_n| \quad (2.4)$$

Mahalanobis distance:

$$r^2 = (x - \mu)^T C^{-1} (x - \mu) \quad (2.5)$$

where $\boldsymbol{C}$ is the covariance matrix of the vector $\boldsymbol{x}$.

An alternative approach may be based on measuring the similarity among two points of the features space through the inner product:

$$s = \frac{x^T \mu}{\|x\| \cdot \|\mu\|} = \cos(\theta) \quad (2.6)$$

where θ is the angle among x and μ. In this case, the similarity among the patterns is maximum for $\theta = 0$.

2.1. The V-C Dimension

The V-C dimension (i.e. Vapnik-Chervonenkis) [9] is a measure of classification capacity for a family of functions that compose a patterns classifier. As an example, a XOR classifier is illustrated in Figure 1. In this case, μ_1 and μ_2 are computed using Equation 2.1 and $\boldsymbol{r}$ is obtained by Equation 2.3

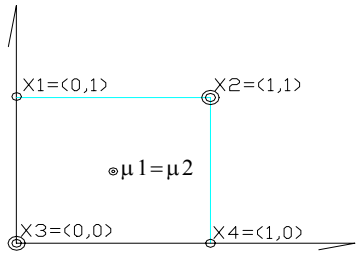

Figure 1. Patterns separation of the XOR problem.

Notice that minimum Euclidean distance classifier has linear decision surfaces. Then, the success of this classifier application is limited to lineally separable patterns. From Figure 1, one concludes that the classification of $\boldsymbol{x_2}$ and $\boldsymbol{x_3}$ as belonging to the pattern defined by $\boldsymbol{\mu_1}$ and also $\boldsymbol{x_1}$ and $\boldsymbol{x_4}$ as belonging to the pattern defined by $\boldsymbol{\mu_2}$ is not possible in the features space $\Re^2$, because the medium vectors of this two patterns coincide. Moreover, if there are only three points to be classified or if the features space possesses three dimensions, it would not be characterized as a classification problem. This problem also occurs when a classical perceptron is used to solve the XOR problem [9].

This kind of restriction can be foreseen through the analysis of the V-C dimension of classifiers. Last example provides a specific case, where the features space has dimension 2, which implicates that its V-C dimension is 3. It means that minimum Euclidean distance classifier is able (i.e. perform classification with probability of mistake equal to zero) to classify any three data in the features space $\Re^2$.

The V-C dimension of a classifier based on a set $\mathfrak{I}$ of Euclidean distance functions or internal product in a features space $\Re^n$ is defined as:

$$\text{VCdim}(\mathfrak{I}) = n+1 \quad (2.7)$$

where $\boldsymbol{n}$ is the number of adjustable parameters (i.e. number of dimensions of the medium vector $\boldsymbol{\mu}$), noticing that, in this case, it is equal to the dimension of the features space.

The V-C dimension indicates how many data can be classified by $\mathfrak{I}$, with percentile of mistake zero. Thus, if the features vector of a minimum Euclidean distance classifier has dimension 10, it is possible to assure the appropriate classification of 11 data. On the other hand, classifiers as multi-layer feedforward ANN with sigmoid transfer function in the hidden layer and linear neurons in the output layer, has V-C dimension computed as:

$$\text{VCdim}(\mathfrak{I}) = k.\boldsymbol{n}^2 \quad (2.8)$$

where $k = cte$ and noticing that the number of adjustable parameters ($\boldsymbol{n}$) is associated to the ANN weights and biases. See [21] for details.

2.2. Statistical Approach Restrictions

The determination of the relevant properties that will compose the features vector, usually demands a high knowledge of the application problem because these features are strongly dependent of the specific problem. Thus, to the same set of data, one can extract different features vectors based on the type of classification task required. As an example, one can mention *recognition of phonemes* and *recognition of announcer* tasks. Both receive the same set of data, however, the features vectors adopted should be different. One consider that the features extraction through a specialist is more an art than a science.

An inadequate choice of the features vector can decrease the classifier performance. Some common problems that occurs due to an inadequate features extraction can be mentioned:

- Correlated features: in this case the information contained in the vector of features may be redundant, because some of their coordinates can be approximated by a linear combination of the other ones. Thus, the amount of information contained in this vector may be insufficient to data appropriated characterization.
- Features in inadequate scales: features relevance may mask the solution. Moreover, numerical problems may occur.
- Feature vectors that will require non-linear decision surfaces for the pattern classification. It means that a more complex classifier (i.e. higher V-C dimension) will be also required.
- Inadequate features: the features are, simply, inadequate or insufficient and they are not able to pattern differentiation.

Quadratic decision surfaces as proposed in the classifiers based on the Mahalanobis distance (see Equation 2.5) are more capable than minimum Euclidean distance for patterns separation. However, the computational cost of the covariance inverse matrix increases proportionally to the

square of the features vector dimension. Therefore, it becomes unviable for many practical applications.
Comparison among equations (2.7) and (2.8) suggests the advantage of the ANN feedforward approaches.

3. ANN FEEDFORWARD MULTI-LAYER IN PR

ANNs feedforward multi-layer are efficient classifiers, due its capacity to produce complex decision surfaces. Such fact is ratified by the analysis of its V-C dimension, which is proportional to the square of the numbers of free parameters (i.e. synaptic weights and biases) [21].

In general, for a ***m*** classes classification, an ANN is used with ***m*** neurons in the output layer. Moreover, the ANN is commonly trained based on binary target data. In these cases the target output vector has all their coordinates set as null, except the one that indexes the class, which the input data belongs. This approach allows a better theoretical analysis, due to facilitating the computation of the V-C dimension.

The use of sigmoid transfer function in the hidden layer and linear in the output layers is usual and also convenient. Notice that this configuration allows analogies with statistical classifiers, as shown further on.

The bayesian rule is the more usual decision rule for the output classification, in other words, the appropriate class is indexed by the largest coordinate of the output vector. Infinite arithmetic precision is considered such that ties are not possible.

Figure 2 describes the ANN as proposed. It is characterized by two matrices: $\boldsymbol{W_1} \in \Re^n \times \Re^j$ e $\boldsymbol{W_2} \in \Re^j \times \Re^m$ where $\boldsymbol{W_1}$ and $\boldsymbol{W_2}$ are the weights matrices.

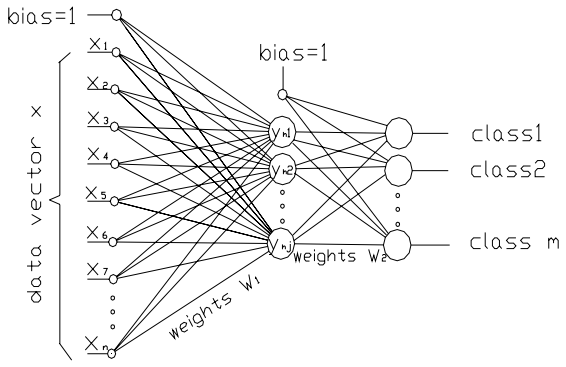

Figure 2. Flowchart of the selected ANN

Bias $\boldsymbol{b_1}$ and $\boldsymbol{b_2}$ is also considered, such that:

$$y_h = \varphi(x \cdot W_1 + b_1) \quad (3.1)$$

$$y = y_h \cdot W_2 + b_2 \quad (3.2)$$

were $\varphi(.)$ is the sigmoid function and y_h is the output vector of the hidden layer.

An interesting approach to the analysis of the ANN behavior in patterns classification, is to associate the hidden neurons as the features extractors and the output neurons (which are linear), to statistical classifiers based on the largest internal product criterion. Thus, the hidden neurons will extract features which characterize the input data, through their non-linear transformation (i.e. Equation 3.1). It transformation defines a new space called as occult space or features space (i.e. characterized by the output vector of the hidden layer $\boldsymbol{y_h}$). In this space, when pattern recognition is achieved, it means that the feature vectors $\boldsymbol{y_h}$ are lineally separable by the output neurons provided by the internal product illustrated in the Equation 3.2.

3.1 Invariability Incorporation in ANN for Pr

A fundamental requirement to the recognition of patterns is the invariability of the features vector with respect to possible transformations in the input data. Such transformations can be exemplified by rotations, shift or scale change for a same set of data.

There are several forms to increasing the ANN recognition robustness with respect to the transformations. Common procedures can be highlighted:

· By the ANN structure. It is obtained through restrictions in the project for incorporation of previous knowledge regarding the task to be accomplished. Convolutive ANN may be applied [17].

· By the training. The ANN is trained by several examples of the same pattern, which contains the pattern transformations. Computational cost of this approach is usually very high.

· By the designing of invariant features extractor. This approach is applicable for any classifier, statistical or connectionist. Previous knowledge regarding to the problem to be treated is required, thus the higher cost of this resource belongs to the planner.

4. SELF ORGANIZING MAP IN PR

The Self-Organizing Map (SOM) [7, 16] is a non-supervised training artificial neural network. Each output neuron represents a single class. An input data activates an only of these neurons.

The main idea of this model is the competitive learning. It means that when input data is given, the neurons compete amongst themselves and the winner weights are adjusted in order to increase its representation of the input signal. The algorithm also provides a cooperation process among the winner neuron and its neighboring neurons, which also have the adjustments of their weights. Therefore, input signal features will stimulate some specific SOM region around the winner neuron. This approach allows us to classify the SOM as a topological paradigm [6].

The motivation for the SOM model creation is the theory that the human brain has different sensorial inputs mapped in specific areas of the cerebral cortex. It is may be denominated as "*probability distribution codified by location*" [3, 6].

Figure 3 illustrates a four neuron SOM from which the input data vector has dimension eight and each SOM neuron is totally connected to the input nodes.

The organizer process begins arbitrating small random values to the weights, usually based on a uniform probability distribution, so that no previous organization is imposed to the map.

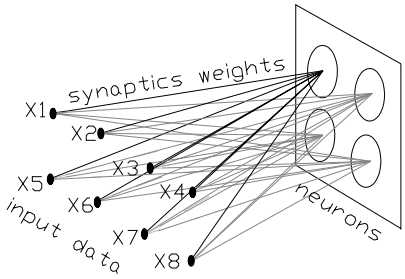

Figure 3. Flowchart of a SOM

Considering Ω as a set of data, an input vector represented by $\boldsymbol{x=[x_1,x_2, ...,x_n]^T}$ is randomly selected from Ω and presented to the net. A single neuron should be better activated by the input data. An usual criterion to the winners choice may be based on Euclidean distance $\boldsymbol{d_{ix}}$ which represents the distance among the input vector $\boldsymbol{x}$ and the synaptic weights of the $\boldsymbol{i^{th}}$ neuron.

Previous work [4] demonstrates that the interaction among a biological neuron and its neighborhood decreases with the increase of the distance. The same property is used by the SOM neuron and a topologic neighborhood parameter $\boldsymbol{h_{ij}}$ is used. It indicates the relationship among the winner neuron $\boldsymbol{i}$ and the $\boldsymbol{j}$ neuron. The $\boldsymbol{h_{ij}}$ parameter is symmetrical in relation to the neuron $\boldsymbol{i}$ and monotonically decreases with the increase of the $\boldsymbol{d_{ij}}$ distance. Also:

$$\lim_{d_{ij}\to\infty} h_{ij} = 0 \qquad (4.1)$$

Therefore, gaussian function is commonly used:

$$h_{ij} = e^{(-\frac{d_{ij}^2}{2\sigma^2})} \qquad (4.2)$$

where $0 \le \boldsymbol{\sigma}$ is considered as the effective width of the topologic neighborhood.

In order to provide neuron specialization (i.e. restricted neighborhood), $\boldsymbol{\sigma}$ may decrease during the iterations evolution. Exponential function are commonly used:

$$\sigma(n) = \sigma_0 \cdot e^{(-\frac{n}{\tau})} \qquad (4.3)$$

where $\boldsymbol{\sigma_0}$ is the initial value of $\boldsymbol{\sigma}$, $\boldsymbol{n}$ is the number of iterations and $\boldsymbol{\tau}$ is a constant.

The SOM learning happens by the adjustment of its synaptic weights $\boldsymbol{w_{ij}}$ among the input node $\boldsymbol{j}$ and the neuron $\boldsymbol{i}$. Considering $\boldsymbol{k}$ as the winner neuron, learning are governed by the expression:

$$\Delta w_{ij} = \eta \cdot h_{ik}(n) \cdot (x_j - w_{ij}) \qquad (4.4)$$

where $0 \le \eta \le 1$ is a learning rate. Dynamic learning rate η(n) can be used in similar way as proposed in Equation 4.3.

Supervised training may refine trained SOM performance [9]. This technique is known as *vector quantization by learning*. It uses previous knowledge of classification such that it moves the winner neuron weights $\boldsymbol{w}$ towards to the input vector $\boldsymbol{x}$ if a correct classification is achieved, otherwise, the weight vector is moved away. Notice that none of other neurons have their weights adjusted.

$w(n+1) = w(n) + \alpha \cdot [x(n) - w(n)]$, if correct classification.

$w(n+1) = w(n) - \alpha \cdot [x(n) - w(n)]$, if incorrect classification.

where $0 < \alpha < 1$.

4.1. SOMs properties and limitations in PR

SOM is usually employed to non-labeled data classification [4]. In this case, it presents interesting properties such as:

· Approximation of the input space $X \in \Re^j$ to the discrete output space $A \in \Re^\gamma$, associated to the neuron coordinates. Since the output neurons layer has low dimension γ (i.e. usually γ= 2), it provides data compression.

· Topologic ordination of data. It is reached because the space location of the neurons is associated to an input features data set and the neighborhood corresponds to similar classes.

· Association of the probability density properties. Areas on the space of input X with larger probability density are mapped in larger domains of the output space A, thus it has more associated neurons.

Based on the presented properties, it is possible to conclude that the vector of weights $\boldsymbol{w_i}$ associated to the neuron $\boldsymbol{i}$, corresponds, in an analogy with the statistical approach, to a prototype $\boldsymbol{\mu_i}$, that characterizes a certain class (see Equation 2.1). However, differently of the statistical approach, the SOM does not request planner knowledge of classes and its relationship with the input data. The SOM will distribute the input data among classes according to the foregoing criteria of density match and topologic ordination.

On the other hand, SOM approach is not justified to be separately used for practical problems of PR that are based on training sets that contains input/output data pairs. In these applications, the properties that differentiate the SOM of the statistical approach (i.e. the capacity to classify data without output-objective) are not explored and the SOM approach is equaled to a statistical classifier based on Euclidean smallest distance. Therefore the SOM classification capacity is limited to lineally separable patterns and its V-C dimension is equal to the statistical classifier V-C dimension (see Equation 2.7).

5. PROPOSED PRÉ-PROCESSING FOR FEATURES EXTRACTION

This work suggests some techniques to refute low relevance data. The main idea is to do not consider data that contain low information level with respect to a pre-defined threshold. Information level will be analyzed based on its entropy.

The central idea is to truncate the input data vector $\boldsymbol{x}$. The new vector should keep enough information such that appropriate data characterization is still possible.

It is important to notice that correlation methods may also be used. However, the calculation of the correlation matrix has a high computational cost that grows with the square of the coordinates number of the data vector. On the other hand, the entropy vector has the same dimension of the

vector of data. Therefore, the entropy verification of the data seems to be the most economical approach.

5.1. Discrete approach

If considered that features are represented by a constraint set of discrete values, it is possible to associate each feature to a discrete random variable.

Computation of the entropy of a discrete random variable ***X*** request the computation of the amount of information ***I*** revealed after occurrence of the event $X=x_i$. Where ***I*** is related to the x_i occurrence rarity. It means that observation of expected events brings low information level. On the other hand, a rare event is surrounded by a very specifics circumstance, which brings new information.

Consider probability is p_i as the probability of the event $X=x_i$ occurrence, then the amount of information is defined as:

$$I(x_i) = \log(\frac{1}{p_i}) = -\log(p_i) \tag{5.1}$$

Notice that the inverse relationship with the probability denotes the notion of rarity. Since the scale is logarithmic, if $p_i=1$ then $I(x_i)=0$ and means that events that are 100% previsible does not contain any new information.

Considering ***N*** possible values of x_i that ***X*** can assume, the entropy is computed as

$$H(X) = E\left[I(x_i)\right] = \sum_{i=1}^{N} -\log(p_i)\cdot p_i \tag{5.2}$$

where E[.] is the expectation statistical operator.

5.2. Continuous approach

If considered that features are represented by a set of continuous values, the appropriate tool is the differential entropy ***h,*** which considers the difference among variables entropy. In this case, the probability of the event $X=x_i$, being ***X*** a continuous random variable with density of probability $f(x)$, is:

$$P\{X = x\} = f(x)\cdot dx \tag{5.3}$$

And the amount of information associated to this event is:

$$I(x) = -\log(f(x)\cdot dx) \tag{5.4}$$

Thus, entropy ***h'*** is the mean value of ***I*** based on all the values that the continuous random variable ***X*** can assume:

$$h'(X) = E[I(x)] = \int_{x=-\infty}^{\infty} -\log(f(x)\cdot dx)\cdot f(x)\cdot dx \tag{5.5}$$

Notice that since ***X*** is a random continuous variable, it can assumes infinites values and implicates that $h'(X)\rightarrow\infty$ because the probability a specific event $X=x_i$ tends to zero.

From Equation 5.5:

$$h'(X) = -\int_{x=-\infty}^{\infty}\log(f(x))\cdot f(x)\cdot dx - \int_{x=-\infty}^{\infty}\log(dx)\cdot f(x)\cdot dx \tag{5.6}$$

Notice that the expression:

$$\int_{x=-\infty}^{\infty}\log(dx)\cdot f(x)\cdot dx \tag{5.7}$$

is common to all the features, thus it is cancelled when a differential approach is adopted. Therefore, ***h*** is defined as:

$$h(X) = -\int_{x=-\infty}^{\infty}\log(f(x))\cdot f(x)\cdot dx \tag{5.8}$$

5.3. Entropy application to dimension reduction

The reduction of the dimension provided by the entropy criterion, may help a correlation approach, since it reduce the computational effort to obtain the correlation matrix. Moreover, the analysis of correlation matrix allows the detection of correlated data pairs that contain redundant information.

Notice that truncation process will only be performed without loss of data if correlation data is equal to 1 or its entropy is null. Rejection of low entropy data is then considered as an adequate way of data pré-processing due to its low computational effort.

It is well known that there are no *mathematical tricks* that can supply inexistent information. Thus planner knowledge is usually requested because is not commonly proved that the training set contains enough information. Moreover, even considering that it contains, the computational effort would be prohibitive when consider the training set dimension.

6. A HYBRID APPROACH TO PR

This work illustrate the use of the SOM associated to a statistical classifier. The basic idea it to apply a non-supervised training in order to achieve feature extraction and then proceed a supervised training according Equation 2.1.

The SOM is used as a features extractor, due to its inherent properties as presented in Section 4.1. Features of the input data will be represented by the coordinates of the winner neuron. Thus, designed SOM should have a larger number of neurons than the number of classes to be recognized. Such neurons are labeled as subclasses of the main classes. The process can be understood as a compression of the input space to a features space or subclasses space.

To proceeds the supervised training, the output vector of the SOM which associates coordinates of the winner neuron, is then classified by a statistical classifier based on the smallest Euclidean distance with respect to the target class.

Thus, each target class is composed by a set of subclasses represented by a respective neuron of the SOM. The decision surfaces are composed by several linear surfaces (i.e. a composition of SOM decision surfaces). The proposed approach provides more complex decision surfaces than a simple statistical classifier or SOM separately working, as illustrated in the figures 4 and 5. Moreover, parameters adjustment for the suggested method is simpler than the adjustment of a feedforward ANN.

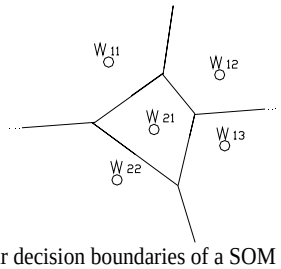

Figure 4. Linear decision boundaries of a SOM features extractor

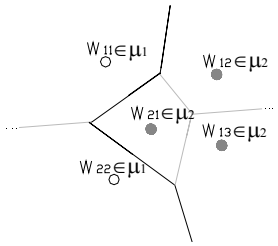

Figure 5. More complex decision boundaries of a hybrid algorithm

7. CONCLUSIONS

To the studied methods in previous sections, the feedforward ANN has the largest patterns separation ability (i.e. V-C dimension). However, if consider that it has a larger number of parameters to be adjusted, training methods should be carefully chosen in order to avoid local minima and also a larger set of training data. Alternative training methods may be applied [1]. The computational effort to ANN training is usually hard.

Even considering the lower V-C dimension of the SOM, it may be successfully applied to features extraction while also presents a low computational cost. In spite of SOM general features extraction ability, it does not refer to the specific classification problem to be treated due to the absence of the target data. Therefore, a hybrid approach is presented.

The presented method seems to be promising. It take advantages of the SOM ability and based on its final supervised training, it becomes able to completely solve classification problems.

Acknowledges:
This work had the support of FAPESB (Brazil) and FCT (Portugal).

REFERENCES

[1] Yasunaga, M.; Nakamura, T.; Yoshimara, I and Kim, J.H.: The kernel-based pattern recognition system designed by genetic algorithm. *IEICE Trans. on Information and Systems*, v E84, n 11, p 1528-1539, Nov 2001.

[2] Lippmann, R.P., *Pattern classification using neural networks*, IEEE Communications Magazine, vol.27, 1989.

[3] Ritter, H., K. Obermayer, K. Schulten.: *Development and spatial structure of cortical feature maps: A model study*, Advances in Neural Information Processing Systems, vol.3, 1991.

[4] von der Malsburg, *Network self-organization*, San Diego, CA: Academic Press, 1990.

[5] Kohonen, T.: *Exploration of very large databases by self-organizing maps*, International Conference on Neural Networks, vol I, 1997.

[6] Kohonen, T.: *Physiological interpretation of the self-organizing map algorithm*, Neural Networs, vol.6, 1993.

[7] Kohonen, T.: *Self organizing maps*, Berlin: Springer-Verlag, 1997.

[8] Haykin, S. et al.: *Classification of radar clutter in air traffic control environment*, Proceedings of the IEEE, vol.79, 1991.

[9] Haykin, S.: *Redes Neurais, princípios e práticas*. Bookman

[10] Parbhane RV, Tambe SS, Kulkarni BD. ANN modelling of DNA sequences: new strategies using DNA shape code. Comput Chem 2000; 24: 699-711.

[11] S.N. Srihari, High-Performance Reading Machines, *Proceedings of the IEEE*, 80(7), July 1992, 1120-1132.

[12] C. C. Yang, M. M. Marefat, and R. L. Kashyap, "Automated Visual Inspecion Based on CAD Models," Proceedings of IEEE International Conference on Robotics and Automation, San Diego, CA, May 8-13, 1994.

[13] Information Visualization in Data Mining and Knowledge Discovery (Morgan Kaufmann), 2001

[14] Carreira-Perpinan, M. A.: *Compression neural networks for feature extraction: Application to human recognition from ear images*. MSc thesis, Faculty of Informatics, Technical University of Madrid, Spain, 1995.

[15] Kim, S.S.; Lee, D.J.; Kwak, K.C.; Park, J.H. and Ryu, J.W.: Speech recognition using integra-normalizer and neuro-fuzzy method. *IEEE Conf. on Signals, Systems and Computers*, v 2, p 1498-1501, 2000.

[16] Teuvo Kohonen; Helge Ritter: *Biological Cybernetics*, 61, 241-254, Elsevier,Amsterdam,1989.

[17] LeCun, Y. and Y. Bengio, *Convolutional Networks for Images, Speech and Time Series*, MIT Press, Cambridge, 1995.

[18] P. A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*, Prentice-Hall International, Englewood Cliffs, NJ, 1980.

[19] M. Stefik, *Introduction to Knowledge Systems*, Morgan Kaufmann, San Francisco, CA, 1995.

[20] K. Fukunaga, *Introduction to Statistical Pattern Recognition, 2nd Ed.*, Academic Press, New York, 1990.

[21] Koiran, P., Sontag, E. D., *Neural Networks with quadratic VC dimension*, Journal of Computer and System Sciences, 54(1):190-198, 1997.