In press in: Neural Networks in Biomedicine
F. Masulli, P. Morasso, and A. Schenone, eds.
World Scientific - Singapore, 1994

BAYESIAN CLASSIFICATION BY FEEDFORWARD CONNECTIONIST SYSTEMS

Francesco Masulli

Department of Physics
University of Genoa
Via Dodecaneso 33 - 16146 Genova, Italy
E-Mail: masulli@genova.infn.it

ABSTRACT

In this paper, some elements of statistical pattern recognition theory are introduced. The approximation of Bayes optimal discriminating function by classifiers based on feedforward connectionist systems, is described. Some classifiers based on connectionist systems, namely the multilayer perceptron (MLP), the resource allocating network (RAN), and the adaptive fuzzy system (AFS), are compared in an application to handwritten character recognition. Theoretical and experimental results are very promising, and permit one to face pattern recognition problems by means of connectionist systems with confidence of success.

1. Introduction

Pattern recognition or *classification* concerns the problem of assigning entities, or *patterns*, to some *classes* or states of nature.

In accordance with the specific classification problem addressed, one can face pattern recognition from a statistical point of view (*statistical pattern recognition*) or a structural one (*structural pattern recognition*). In statistical pattern recognition, a pattern is regarded as a stochastic vector and statistical methods are used for classification. On the contrary, in structural pattern recognition one assumes that the structure of a pattern is of paramount importance and can be the basis for the pattern description and classification, e.g. using a grammar that generates the specific structure of each pattern class[36] .

Human beings, and other biological beings can accomplish some pattern recognition tasks (e.g. face recognition, handwritten document reading, speech recognition, etc.) in a efficient and immediate way. Automation of pattern recognition tasks has many advantages in terms of speed-up and standardization. Moreover, it allows one to face new tasks that are difficult to human beings, e.g. when some inputs cannot be detected by the human sense-organs, or a-priori knowledge is not available or too complex. Such tasks include long-term weather forecast, protein secondary structure prediction, medical diagnosis, and so on.

Artificial neural networks, as mathematical metaphors of biological nervous systems, have aroused great interest in the area of pattern recognition since the 40's, when they were first proposed[25,32,38] , even though their applications to real cases have often been unsatisfactory. In particular, only in the '80 were some significant theoretical results obtained, which made it possible to overcome the linear separability problem[26,35] . At present, many theoretical results supporting the applications of

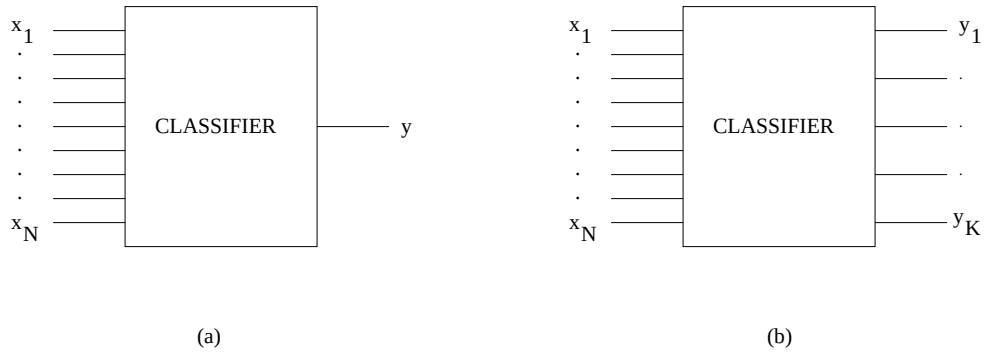

Figure 1: *Types of classifiers.*

neural networks and of other connectionist systems are available[6] , such as those concerning the learning methods in multilayer networks[34] , function approximation[3] , and approximation of the Bayes discriminant function[33, 9, 16, 27, 1] . Such results, combined with the diffusion of high-performance computers, permit one to face even complex pattern recognition problems by means of connectionist systems, with confidence of success.

The purposes of this paper are to introduce some basic elements of statistical pattern recognition, to describe some some classifiers based on feedforward connectionist systems, namely the multilayer perceptron[35] (MLP), the resource allocating network[30] (RAN), and the adaptive fuzzy system[15, 23, 2] (AFS), and to compare the performances of such systems in a specific application, that is, automatic handwritten character recognition.

For a deeper analysis of the the topics dealt with in this paper we refer the reader to the books by Duda and Hart[5] , Fukunaga[7] , and Therrien[37] on pattern recognition, to those by Hertz, Krogh, and Palmer[11] , and Kosko[20] on neural networks and fuzzy systems.

This article is organized as follows. After this introduction, Section 2 presents the basic elements of pattern recognition. In Section 3 neural networks and connectionist systems are introduced. In Section 4 the approximation of the Bayes optimal discriminant function by classifiers based on feedforward connectionist systems is discussed. In Section 5, the performances of some connectionist systems in handwritten characters recognition, are compared. Finally, in Section 6 the conclusions are drawn. An Appendix on similarity measures used in pattern recognition completes the article.

2. Statistical Pattern Recognition

2.1. Patterns and Classifiers

In statistical pattern recognition, an entity, or *pattern*, to be classified, is regarded as a random vector

$$\mathbf{x} = (x_i \mid x_i \in \Re,\ \ i \in [1, I]), \tag{1}$$

where the components x_i are also named the pattern *features*.

Let us denote with C the set of K *classes* to which the pattern might belong:

$$C = \{c_k \mid k \in [1, K]\}. \tag{2}$$

A *classifier* can be regarded as a black box capable to associate an input pattern with a certain class. In Figure 1 two possible types of classifier are shown[28] . For the first type (Figure 1a), the output is a one-dimensional variable that assumes integer values dependent on the class associated by the classifier to the input pattern. For the other case (Figure 1b), the output $\mathbf{y}$ is a label represented by a binary vector whose elements are all zero, with the exception of the one set to 1, that is associated with the class to which the input pattern belongs:

$$\mathbf{y} = (y_k \mid k \in [1, K]) \tag{3}$$

$$y_k = \begin{cases} 1 & \text{if the example belongs to the class } c_k \ , \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

2.2. Supervised Pattern Recognition

In *supervised pattern recognition* the classifier is designed by using a data-base of N *associative pairs*

$$(\mathbf{x}^n, \mathbf{t}^n), \quad n \in [1, N] \tag{5}$$

i.e., a set of pairs, each made up of a specimen pattern $\mathbf{x}^n$ and of a label $\mathbf{t}^n$ denoting the class the pattern belongs to.

The classifier performance depends on the statistical significance of the used set of associative pairs with respect to the whole pattern population.

The design of a supervised classifier generally involves two phases:

- In the first phase, i.e. the *training phase*, a subset of available associative pairs (called the *training set*) is used to construct the classifier. Usually, this phase consists in tuning the classifier parameters or the classifier structure for the purpose of obtaining a correct working of the classifier on the training set.
- In the second phase, i.e. the *test phase*, another subset of associative pairs (called the *test set*) is used, that is independent of the training set, thus allowing one to study the *generalization* capability of the classifier, i.e., how it works on patterns not used in the training phase.

It is worth to note that is useful to provide a *rejection class* (or *null class*), which the classifier could utilize when it can't associate a pattern to a known class with a sufficient confidence level.

Typical problems where supervised pattern recognition methods are successfully applied are handwriting recognition, medical pathology recognition, biological cell recognition, and so on.

2.3. Unsupervised Pattern Recognition

Unsupervised pattern recognition or *clustering* is used when a data-base of associative pairs is not available and only a set of unlabeled pattern examples can be utilized as a-priori knowledge related to the addressed problem.

Unsupervised methods are applied in cases where:

- the kind or number of classes to which the pattern might belong are unknown;
- the labeling operation is slow or inaccurate;

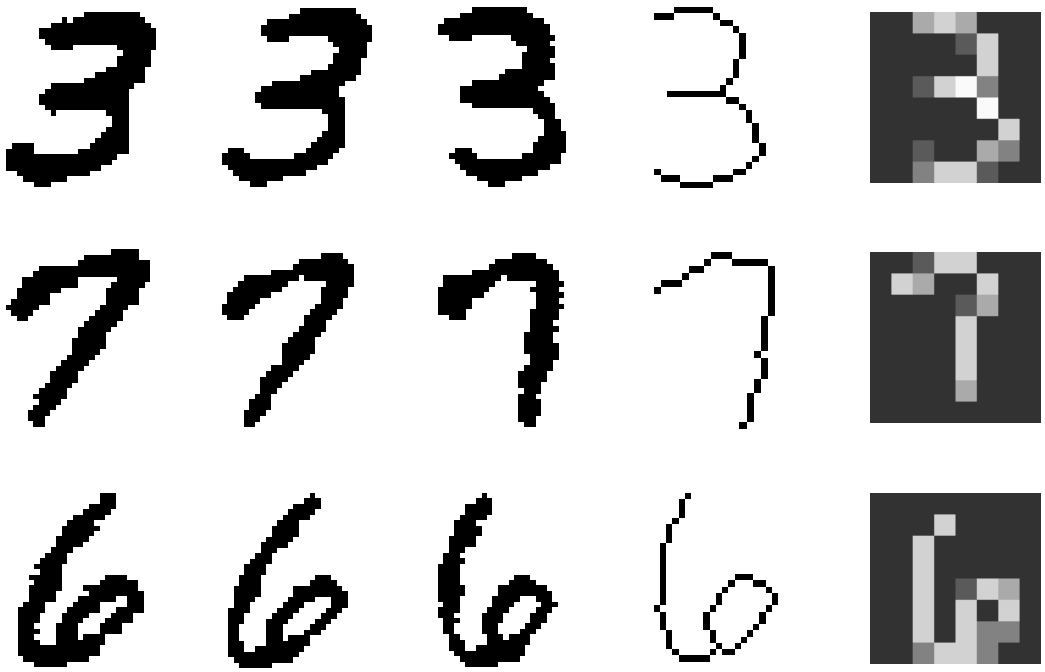

Figure 2: Character preprocessing.

- a-priori knowledge is ambiguous, as in bio-medicine, where a pattern may be labeled in different ways by different human experts[4] .

In unsupervised classification, the classification system tries to organize the training set of unlabeled patterns into some clusters by using a specific similarity measure. In the Appendix, some of the most used similarity measures are shown.

After the clustering process, the a-priori knowledge can be used to label the resulting clusters. Typical problems were unsupervised pattern recognition techniques are used are image texture segmentation, biomedical image understanding, speech recognition, and so on.

2.4. Features

In order to facilitate the classifier's task, the patterns are generally preprocessed before being classified, through a transformation from the measure space to the space of so-called *features*. The goal of the preprocessing procedure is to obtain a new pattern representation that reduces pattern variability and redundancy[5] . As examples of preprocessing steps, we recall noise filtering, the study of occlusions, distortions, symmetries and scaling, sampling methods and principal component analysis[37] .

Preprocessing is an empirical task and depends on the specific classification problem addressed. It is impossible to give a general criterion for feature choice; moreover, some transformations (such as the symmetry detection) may be useful in some cases and dangerous in others.

Here we describe the preprocessing procedure used for the comparison of connectionist classifiers reported in Section 5.. For this comparison, the training and test

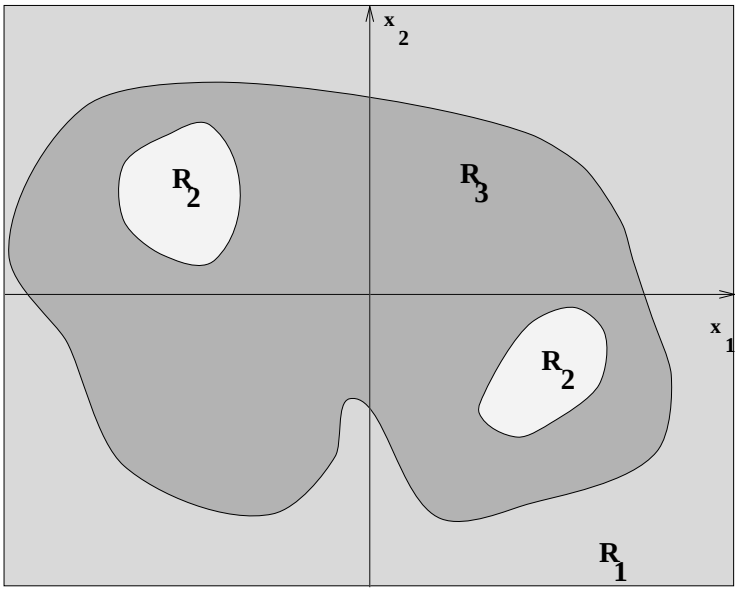

Figure 3: *An example of decision regions.*

sets, each set containing 10,000 associative pairs of segmented handwritten characters, were extracted from the NIST-3[8] data-base.

The NIST-3 data-base, distributed on a standard cd-rom, contains 313389 characters coded as 128×128 matrix binary images and labeled by the corresponding ASCII codes. As shown in Figure 2, the preprocessing required the following steps:

- a character image was extracted from the cd-rom and normalized to a 32×32 binary matrix (Figure 2a);
- a low-pass filter was applied in order to remove some small spots and holes in the image (Figure 2b);
- a shear transform was applied to the character image to straighten the axis joining the first upper-left point of the character image to the last lower-right point (Figure 2c);
- the image was then skeletonized by using a thinning algorithm[29] (Figure 2d);
- finally, the character representation was transformed into a 64-element vector; each vector element represented the number of black pixels contained in adjacent 4×4 squares (Figure 2e).

It is worth pointing out that the obtained character representation presents a sufficient degree of invariance both to the scale and to small image shifts or rotations.

2.5. Elements of Decision Theory

The purpose of pattern recognition is to separate regions in the feature spaces containing patterns belonging to the same class or cluster.

To this end, a classifier exploits the a-priori knowledge contained in the pattern labels of the training set for supervised pattern recognition, or uses an assigned similarity measure for unsupervised pattern recognition.

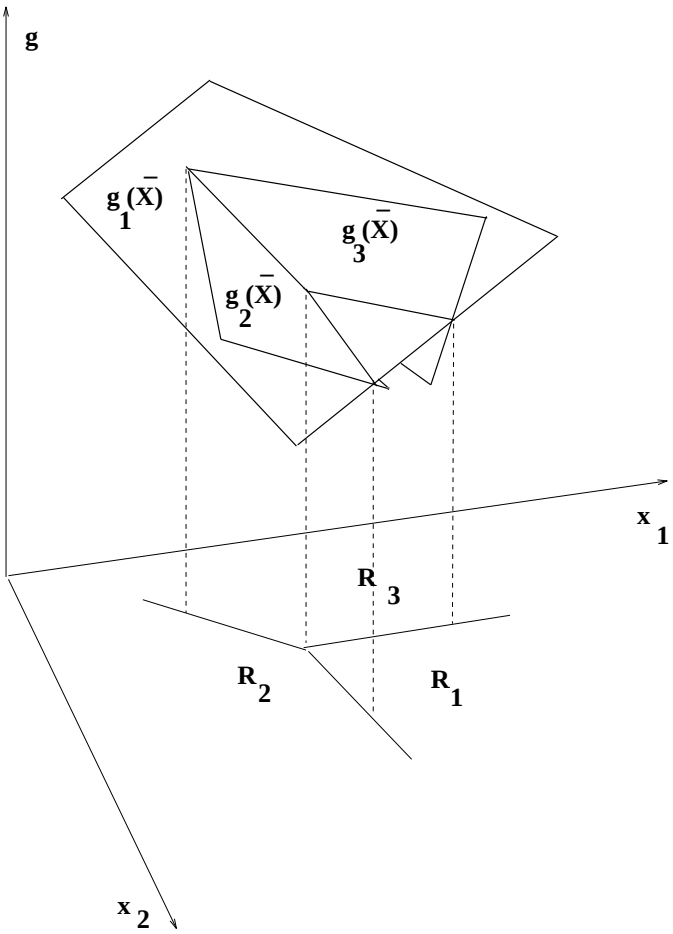

Figure 4: *Discriminant functions, decision regions, and decision surfaces in a three-classes and two-features case.*

The decision on assigning a pattern to a class can be made by using suitable *decision rules*, or an equivalent decision criterion consists in associating with each class c_j a scalar function $g_j(\mathbf{x})$, called the *discriminant function*, and in assigning an unknown pattern to the class for which the discriminant function assumes the highest value for that pattern:

$$\mathbf{x} \in c_j \;\Leftrightarrow\; g_j(\mathbf{x}) > g_l(\mathbf{x}) \quad j,l \in [1,K] \;\; j \neq l. \tag{6}$$

The discriminant function is determined by using the training set.

A further decision criterion, equivalent to the previous ones, is obtained by determining the so-called *decision regions* R_k, which are the regions that the classifier associates with different classes (see Figure 3). A decision region corresponds to the subspace where the discriminant function for a given pattern has the highest value:

$$R_i = \{\mathbf{x} \mid g_i(\mathbf{x}) \geq g_j(\mathbf{x}) \quad i,j \in [1,K] \quad i \neq j\} \tag{7}$$

Decision surfaces are the surfaces separating adjacent decision regions, e.g.:

$$g_i(\mathbf{x}) = g_j(\mathbf{x}) \qquad \text{for } R_i \text{ and } R_j. \tag{8}$$

Figure 4 shows a simple geometrical sketch of discriminant functions, decision regions, and decision surfaces in a three-classes and two-features case.

2.6. Bayes Decision Rule

Let we assume we to want to realize a system able to assign some entities to two independent classes, c_A e c_B. Let $P(c_A)$, and $P(c_B)$ be the *a-priori probabilities* (or *"a-priori"*) of the two classes, respectively.

If the measures of the entities to be classified are not available, the optimal classification criterion, i.e. the one giving the minimum probability of classification error (or *misclassification error*), is to select the class with the highest a-priori probability.

On the contrary, if the measures of the patterns are available we can define $P(c_k \mid \mathbf{x}^*)$ as the conditional probability that the class may be k when the pattern is $\mathbf{x}^*$. This probability is called *a-posteriori probability* (or *"a-posteriori"*).

The decision criterion giving the minimum probability of classification error is the *Bayes optimal decision rule*, which assumes as a discriminant function (*Bayes discriminant function*) the a-posteriori class probability, i.e.,

$$g_k(\mathbf{x}) = P(c_k \mid \mathbf{x}). \tag{9}$$

Therefore the Bayes optimal decision rule is:

$$\textit{choose class } i \;\Leftrightarrow\; p(c_i \mid \mathbf{x}) > p(c_j \mid \mathbf{x}) \quad \forall i \neq j. \tag{10}$$

The joint probability that the class may be c_k and the pattern is $\mathbf{x}^*$ is defined as:

$$P(c_k, \mathbf{x}^*) = P(c_k \mid \mathbf{x}^*)P(\mathbf{x}^*) \tag{11}$$

as well as

$$P(c_k, \mathbf{x}^*) = P(\mathbf{x}^* \mid c_k)P(c_k) \tag{12}$$

From Eqs (11) and (12), the *Bayes formula* follows immediately, which gives a direct relation between a-priori and a-posteriori probabilities:

$$P(c_k \mid \mathbf{x}^*) = \frac{P(\mathbf{x}^* \mid c_k)P(c_k)}{P(\mathbf{x}^*)}. \tag{13}$$

If $\mathbf{x}$ is a vector with continuous elements values, the Bayes formula becomes:

$$P(c_k \mid \mathbf{x}^*) = \frac{p(\mathbf{x}^* \mid c_k)P(c_k)}{p(\mathbf{x}^*)}. \tag{14}$$

It is worth noting that $p(\mathbf{x}^*)$ is independent of the class and acts as a normalization factor of a-posteriori probabilities. As a consequence, we can use the following Bayes discriminant function:

$$g_k(\mathbf{x}) = p(\mathbf{x} \mid c_k)P(c_k). \tag{15}$$

The new Bayes optimal decision rule is then:

$$\textit{choose class } i \;\Leftrightarrow\; p(\mathbf{x} \mid c_i)P(c_i) > p(\mathbf{x} \mid c_j)P(c_j) \quad \forall i \neq j. \tag{16}$$

In order to show the effectiveness of the Bayes decision criterion, let us consider a classification case using two classes c_1 and c_2 (dicotomizator) and only one feature, x. Figure 5 shows the classes' a-posteriori probabilities, assumed to be known. The

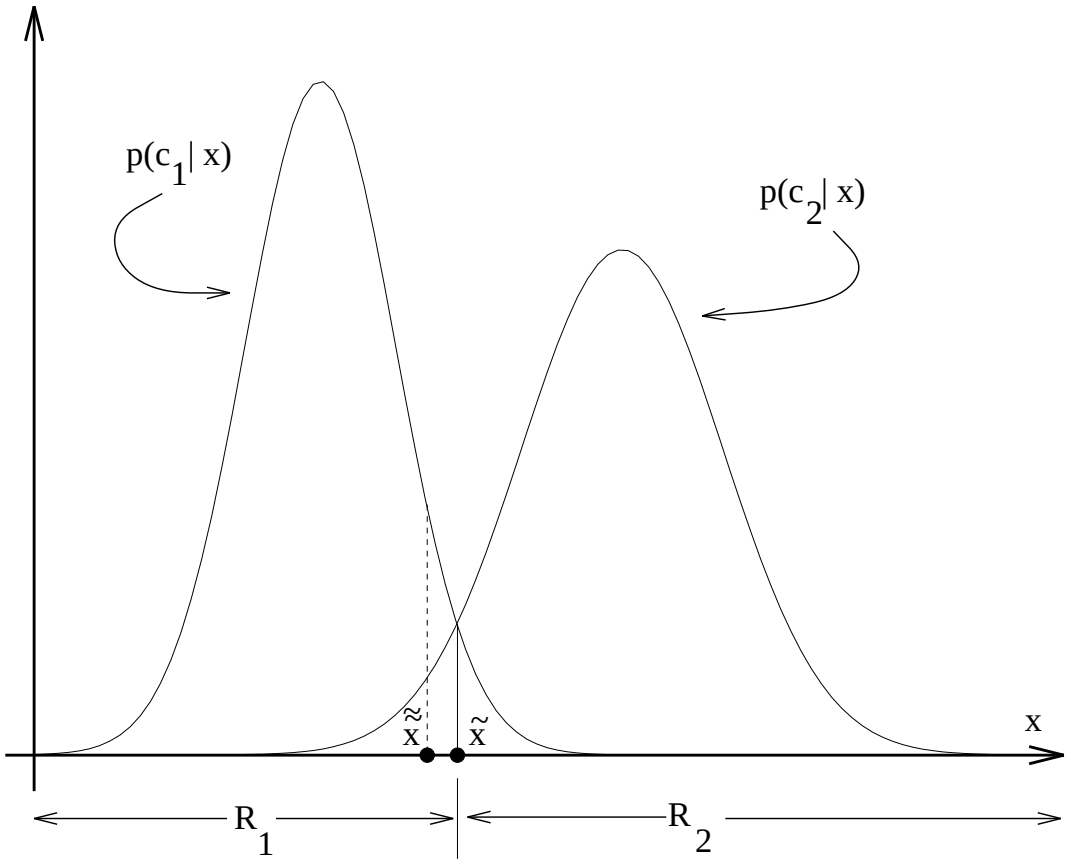

Figure 5: *Bayes decision rule.*

Bayes optimal decision criterion selects as the decision surface the point $\tilde{\mathbf{x}}$. As can be noticed, this choice minimizes the area corresponding to the misclassification probability. In fact, any different choice (e.g. the point $\tilde{\tilde{\mathbf{x}}}$) will give a higher misclassification probability.

We point out that, for the two classes case, we could use the Bayes discriminant function:

$$g(\mathbf{x}) = P(c_1 \mid \mathbf{x}) - P(c_2 \mid \mathbf{x}), \tag{17}$$

and the following Bayes optimal decision criterion:

$$\textit{choose} \;\; c_1 \; \textit{if} \;\; g(\mathbf{x}) \;\; \textit{is positive, otherwise choose} \;\; c_2. \tag{18}$$

2.7. Statistical Pattern Recognition Methods

As mentioned earlier, the Bayes optimal classifier minimizes the misclassification probability. In real cases, it is difficult to use, as the conditional probability distributions $p(\mathbf{x} \mid c_k)$ are seldom known.

Pattern recognition methods, used in both supervised or unsupervised cases, can be divided into:

- *Parametric methods*, used if the forms of $p(\mathbf{x} \mid c_k)$ are known (or assumed to be known) and the probability distribution parameters can be determined so as to

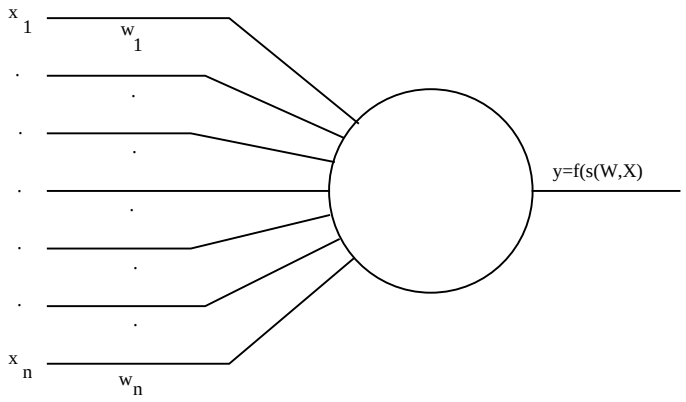

Figure 6: *Basic processor element or neuron.*

minimize the misclassification probability. Classical techniques for parameter determination are the maximum likelihood method and the Bayes estimation methods[5] .

- *Nonparametric methods*, used if even the forms of $p(\mathbf{x} \mid c_k)$ are unknown. Classical nonparametric methods include Parzen windows method[5] and the K-nearest neighbor rule[10] .

An interesting type of nonparametric classifiers is represented by the classifiers that assume the form of the discriminant function as known, and tune the parameters of such a function during the training phase. In the following section, we shall analyze an important class of such type of classifiers, namely neural networks and other connectionist systems able to perform function approximation.

3. Neural Networks and Feedforward Connectionist Systems

In the last few years, promising theoretical and experimental results were obtained for neural networks and connectionist systems. In particular, it has been shown that such systems can perform function approximation[3,31,19,14] , Bayesian classification[22,33,9,16] , and clustering of inputs (unsupervised classification)[18] . Moreover they can be used as content-addressable memories[12,13] .

Artificial neural networks are made up of simple *nodes* or *neurons* interconnected to one another. Generally speaking, a node of a neural network can be regarded as a block that measures the similarity (see the Appendix) between the input vector and the parameter vector, or *weight vector*, associated to the node, followed by another block that computes an activation function, normally not linear. In Figure 6 a basic node is shown, and in Figure 7, some of the most widely used neural activation functions are presented.

In accordance with this scheme, a *perceptron* consists of a single node using the correlation as a similarity measure, and a step function (Figure 7c), a linear function (Figure 7a), or a sigmoid (Figure 7d) as the activation function. The equation for a perceptron is then:

$$y = H(\sum_i w_i x_i - \theta) \tag{19}$$

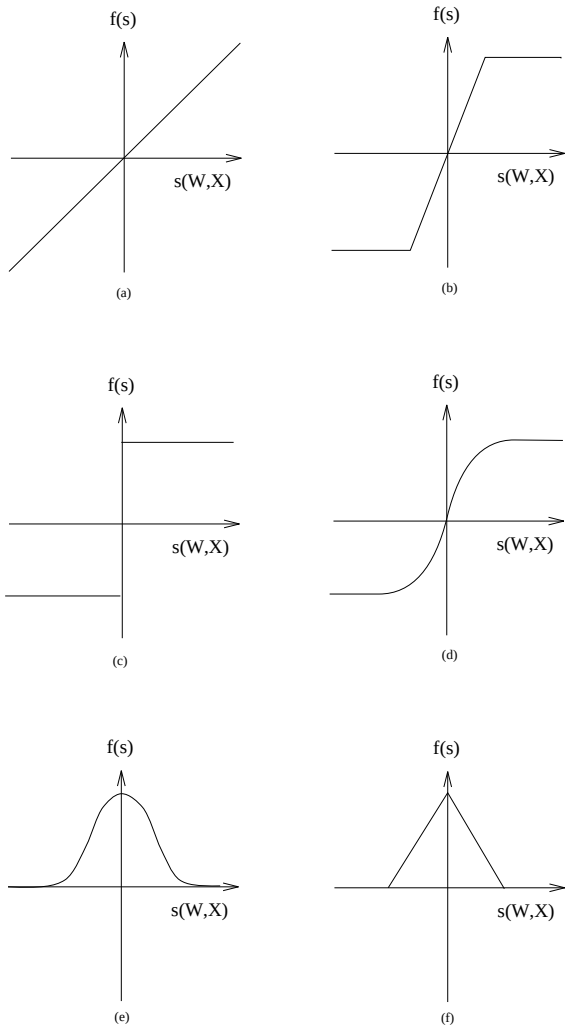

Figure 7: *Activation functions.*

where H is the activation function, w_i are the perceptron weights), and θ is the threshold.

As is well known, a perceptron can discriminate only between classes that can be separated by hyperplanes.[26] This limitation is called the problem of *linear separability.*

Following the same scheme, many neural networks proposed in the literature can be assembled ; moreover, one can realize brand-new neural networks.

For instance, it is possible to obtain the MLP that is a feedforward multilayer networks of perceptrons, or other connectionist feedforward systems such as the RN or the AFS.

In the RN,[31,30] for the units of the hidden layer the similarity measure is the Euclidean distance and the activation function is a *radial basis function* (RBF), e.g. a cone (Figure 7f), a paraboloid or a Gaussian function (Figure 7e):

$$z_j = \exp\left(-\frac{\sum_j (x_i - m_{ji})^2}{\sigma_j^2}\right), \tag{20}$$

where m_{ji} and σ_j are the means and the variances. For the units of the output layer,

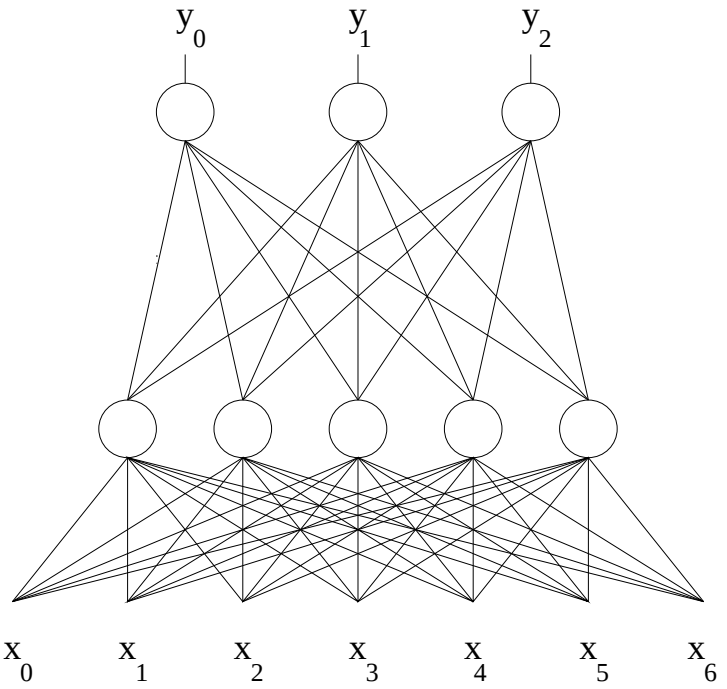

Figure 8: *A feedforward connectionist system.*

the similarity measure is the correlation, and the activation function is linear.

AFSs[15] can be regarded as feedforward connectionist systems with just one hidden layer whose units correspond to the fuzzy rules. The activation of those rules can be written as:

$$r_j \;=\; \prod_i \mu_{ji}(x_i), \tag{21}$$

where the quantity $\mu_{ji}(x_i)$ represents the value of the membership function of the component x_i of the input vector for the j-th rule, and is defined as:

$$\mu_{ji}(x_i) = \exp\left(-\frac{(x_i - m_{ji})^2}{2\sigma_{ji}^2}\right), \tag{22}$$

where m_{ji} and σ_{ji} are the means and the variances. The values of the output units are obtained using a defuzzification rule based on the centroid[20,14] :

$$y_k = \frac{\sum_j r_j s_{kj}}{\sum_j r_j} \;\;, \tag{23}$$

where s_{kj} is the fuzzy singleton of the $j-th$ rule associated to the y_k output.

In such classification systems the training phase consists in the adaptive modification of the system parameters, or *weights*, in order to minimize a cost function that, for instance, can be expressed as the *mean square error* (MSE):

$$MSE = \frac{\sum_{k,n}(y_k^n - t_k^n)^2}{N} \;\;, \tag{24}$$

where $\mathbf{y}^n = (y_k^n)$ is the network output and $\mathbf{t}^n = (t_k^n)$ the n-th label of the associative pair of the training set. The $\mathbf{t}^n$ components can be defined as follows:

$$t_j = \begin{cases} 1 & \text{if the pattern belongs to class } c_j, \\ 0 & \text{otherwise.} \end{cases} \tag{25}$$

The cost function can be minimized by different techniques, among which the gradient descent algorithm is the most popular[11] . For MLPs the gradient descent algorithm is called also *error backpropagation* technique[34] .

It has been shown that MLPs. RN, and AFS are feedforward connectionist systems able to perform function approximation[3,31,19,14] .

Moreover, it has been demonstrated that MLPs can approximate the Bayes optimal discriminant function, for suitable choices of the cost function to be approximated during the training phase, and using a large training set[33,9,16,27,1] .

4. Bayes Optimal Classifier Approximation by Feedforward Connectionist Systems

As pointed out by Ruck et al.[33] , the demonstration that MLPs can approximate the Bayes optimal discriminant function is not based on any assumption about the particular feedforward system used; only the system's capability for function approximation is assumed. In this section, we demonstrate, following Ruch et al.[33] , that a classifier based on a connectionist system able to perform function approximation, can approximate the Bayes discriminant function. The demonstration refers to the two-class case.

Let c_1 e c_2 be two classes of patterns. Let χ_i be the set of all possible patterns $\mathbf{x}$ belonging to class c_i; then $\chi = \chi_1 \cup \chi_2$ represents the set of all patterns. The training set consists of a subset of possible patterns belonging to the two classes. In general, it is a finite set $X = X_1 \cup X_2$, where $X_1 = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{n_1}\} \subset \chi_1$, $X_2 = \{\mathbf{x}_{n_1+1}, ..., \mathbf{x}_{n_1+n_2}\} \subset \chi_2$, and $n_1 + n_2 \equiv N$.

Let $\mathbf{w}$ be the set of adaptive parameters, $g(\mathbf{x})$ be the discriminant function of a Bayes dicotimizator (Eq. (17)), and $F(\mathbf{x}, \mathbf{w})$ be the system output.

Moreover, let us suppose that the system is trained in such a way that its response is $+1$ when $\mathbf{x}$ belongs to the class c_1, and -1 when $\mathbf{x}$ belongs to c_2:

$$F(\mathbf{x}, \mathbf{w}) = \begin{cases} +1 & \text{if } \mathbf{x} \in c_1 \\ -1 & \text{if } \mathbf{x} \in c_2 \end{cases} \tag{26}$$

This is accomplished by mimizing (e.g., with the error backpropagation technique[34]) the *sample data error function*:

$$E_s(\mathbf{w}) = \sum_{\mathbf{x} \in X_1} [F(\mathbf{x}, \mathbf{w}) - 1]^2 \; + \; \sum_{\mathbf{x} \in X_2} [F(\mathbf{x}, \mathbf{w}) + 1]^2 \; . \tag{27}$$

We shall demonstrate that in the large N limit, when $\mathbf{w}$ minimizes $E_s(\mathbf{w})$, then $\mathbf{w}$ minimizes also :

$$\epsilon^2(\mathbf{w}) = \int_{\chi} [F(\mathbf{x}, \mathbf{w}) - g(\mathbf{x})]^2 p(\mathbf{x}) d\mathbf{x}, \tag{28}$$

i.e., $F(\mathbf{x}, \mathbf{w})$ approximates $g(\mathbf{x})$.

Let us define the *average error function* as the function

$$E_a(\mathbf{w}) = \lim_{N\to\infty} \frac{1}{N} E_s(\mathbf{w}) \tag{29}$$

where N is the total number of pattern vectors. $E_a(\mathbf{w})$ represents the error surface that is obtained when all possible vectors are used for the computation.

As assumed, in the large N limit, we can suppose that the function E_s is a reasonable approximation for E_a.

Let us rewrite the function E_a as follows:

$$E_a(\mathbf{w}) = \lim_{N\to\infty} [\frac{n_1}{N} * \frac{1}{n_1} \sum_{\mathbf{x}\in X_1} [F(\mathbf{x},\mathbf{w}) - 1]^2 \; + \; \frac{n_2}{N} * \frac{1}{n_2} \sum_{\mathbf{x}\in X_2} [F(\mathbf{x},\mathbf{w}) + 1]^2] \tag{30}$$

where n_i is the number of vectors belonging to the class c_i, and $\frac{n_1}{N}$ and $\frac{n_2}{N}$ represent, in the large N limit, the a-priori probabilities $P(c_1)$ and $P(c_2)$, respectively.

The quantities $\frac{1}{n_i}\sum_{\mathbf{x}\in X_i}[F(\mathbf{x},\mathbf{w}) - 1]^2$ represent the mean value of $[F(\mathbf{x},\mathbf{w}) - 1]^2$, over the patterns, provided that $\mathbf{x} \in c_i$.

By exploiting the strong law of large number[29] , we can rewrite Eq. (30) as

$$\begin{aligned} E_a(\mathbf{w}) &= P(c_1)\int_\chi [F(\mathbf{x},\mathbf{w}) - 1]^2 p(\mathbf{x} \mid c_1) d\mathbf{x} + \\ &\quad + P(c_2)\int_\chi [F(\mathbf{x},\mathbf{w}) + 1]^2 p(\mathbf{x} \mid c_2) d\mathbf{x} \\ &= \int_\chi [F^2(\mathbf{x},\mathbf{w}) + 1][p(\mathbf{x} \mid c_1)P(c_1) + p(\mathbf{x} \mid c_2)P(c_2)] d\mathbf{x} - \\ &\quad 2\int_\chi F(\mathbf{x},\mathbf{w})[p(\mathbf{x} \mid c_1)P(c_1) - p(\mathbf{x} \mid c_2)P(c_2)] d\mathbf{x} \end{aligned} \tag{31}$$

The probability density function of the input vectors can be expressed as:

$$p(\mathbf{x}) = \; p(\mathbf{x} \mid c_1)P(c_1) \; + \; p(\mathbf{x} \mid c_2)P(c_2) \tag{32}$$

Moreover, by using the Bayes rule (Eq. (14)), we can write:

$$\begin{aligned} g(\mathbf{x})p(\mathbf{x}) &= [P(c_1 \mid \mathbf{x}) - P(c_2 \mid \mathbf{x})]p(\mathbf{x}) \\ &= P(c_1 \mid \mathbf{x})p(\mathbf{x}) - P(c_2 \mid \mathbf{x})p(\mathbf{x}) \\ &= p(\mathbf{x} \mid c_1)P(c_1) - p(\mathbf{x} \mid c_2)P(c_2) \end{aligned} \tag{33}$$

Therefore:

$$\begin{aligned} E_a(\mathbf{w}) &= \int_\chi [F^2(\mathbf{x},\mathbf{w}) + 1]p(\mathbf{x}) d\mathbf{x} - 2\int_\chi F(\mathbf{x},\mathbf{w})g(\mathbf{x})p(\mathbf{x}) \\ &= \int_\chi [F^2(\mathbf{x},\mathbf{w}) - 2F(\mathbf{x},\mathbf{w})g(\mathbf{x})]p(\mathbf{x}) d\mathbf{x} + \int_\chi p(\mathbf{x}) d\mathbf{x} \\ &= \int_\chi [F(\mathbf{x},\mathbf{w}) - g(\mathbf{x})]^2 p(\mathbf{x}) d\mathbf{x} - \int_\chi g^2(\mathbf{x})p(\mathbf{x}) d\mathbf{x} + \int_\chi p(\mathbf{x}) d\mathbf{x} \\ &= \epsilon^2(\mathbf{w}) + \int_\chi [1 - g^2(\mathbf{x})]p(\mathbf{x}) d\mathbf{x} \end{aligned} \tag{34}$$

The learning algorithm minimizes E_s with respect $\mathbf{w}$ and, as we assumed $E_s(\mathbf{w})$ to be a reasonable approximation for $E_a(\mathbf{w})$, this algorithm minimizes also E_a with respect $\mathbf{w}$. Moreover, $\int_\chi [1-g^2(\mathbf{x})]p(\mathbf{x})d\mathbf{x}$ is a quantity that does not depend on $\mathbf{w}$, hence the optimization algorithm minimizes $\epsilon^2(\mathbf{w})$, too, which was to be demonstrated. □

For the multiclass problem, in the large training set dimension limit, one can demonstrate that, if the MSE, defined as in Eq.s (24, 25), is assumed as the cost function, then, when $\mathbf{w}$ minimize the MSE, the system outputs y_k-s approximate the Bayes optimal discriminant functions, i.e. the a-posteriori class probabilities (Eq. (9))[33] .

5. Experimental Comparison of Connectionist Systems

The promising theoretical results reported in the previous section allow one to expect to perform an accurate pattern recognition by using feedforward connectionist systems. On the basis of such results, we carried out an experiment to compare the performances of an MLP, of an RN, and of an AFS, by using for all of them the training and test set described in Subsection 2.4..

All three connectionist system were trained to work as classifiers by minimizing during the training phase the MSE, as defined in Eq.s (24, 25).

In the following, we give some technical details about the implementations of the three connectionist systems:

- The MLP has an input layer of 64 units, a hidden layer of 48 units and an output layer of 10 units. The training algorithm is based on the backpropagation rule[34] accelerated using the *incremental input dimensionality* technique (IID), which is based on the Karunhen-Loève transformation[24] .

- The RN is implemented as a *resource allocating network* (RAN), which is characterized by the growth of its architecture during the training phase[30] . For this system, two learning strategies are available: allocation of new units (RBFs), or changes, by backpropagation technique, in the values of the system parameters. In front of a pattern to be learned, a new unit is allocated if the error of the network is larger than a fixed threshold, and if other units are not present in the neighborhood of the input pattern. At each learning epoch, the threshold for allocation of units is decreased. Our RAN classifier contains 64 inputs and 10 outputs. Typically, during the learning of our training set, about seven hundred hidden units are allocated.

- As explained in Section 3. , the AFS can be viewed as a connectionist feedforward system with just one hidden layer, whose units correspond to fuzzy rules. During the training phase, the system parameters are determined, in an adaptive way, by using the gradient descent technique. The built classifier is an AFS consisting of 64 inputs, 10 outputs and 48 rules.

5.1. Results

As shown in Figure 9, the three nets exhibit similar generalization properties, as we expected from a theoretical standpoint.

The MLP reaches a smaller generalization value than the other two systems. This may be due to the problem of *false positive*, as discussed by Lee.[21] The AFS is very fast during the training phase (10 to 100 times faster than the MLP, which shows the

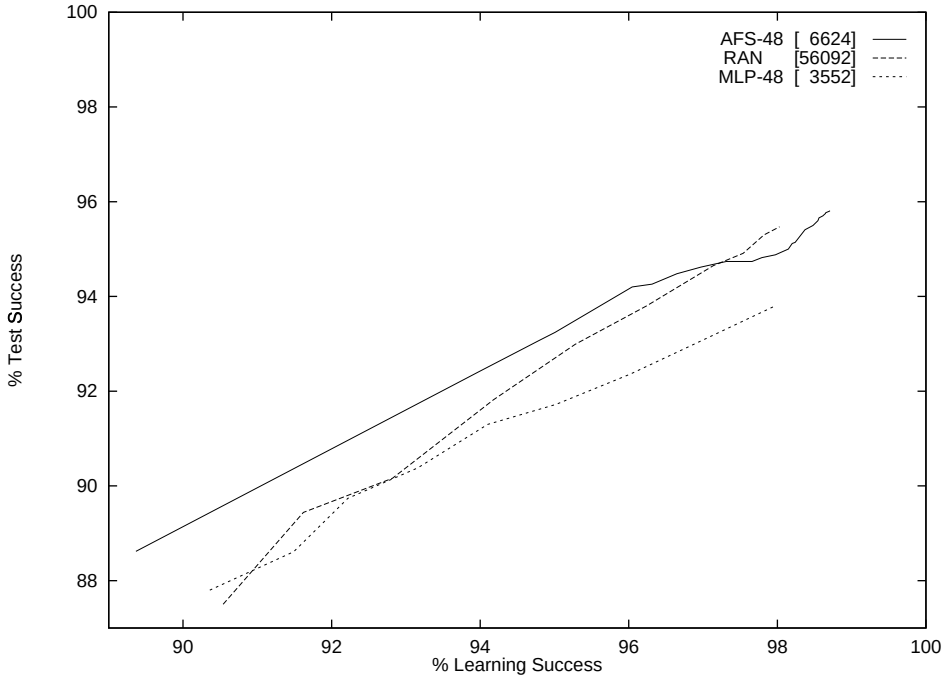

Figure 9: *Comparison among the performances of the MLP using IID, of the AFS and of the RAN. The numbers within brackets refer to the parameters used by each of the three networks.*

same complexity) and converges in a limited number of epochs. On the contrary the RAN is the slowest one during the training phase; this depends on the growth of its architecture up to more than $50,000$ parameters (each of them must be optimized !). Moreover, it is interesting to note that the RAN shows the highest derivative of the Test Success, as compared with the Training Success. This depends on the possibility of allocating new units dynamically during the learning phase for such as system.

6. Conclusions

In this paper, we have studied the use of classifiers based on connectionist system in the framework of statistical pattern recognition, and we have assessed the performances of such classifiers in term of handwritten character recognition.

Theoretical and experimental results are very satisfactory. The feedforward connectionist systems analyzed (MPLs, RNs, and AFSs) allow the realization of non-parametric classification systems able to approximate the Bayes optimal classifier when a large training set is available. Such results allow one to face pattern recognition problems of high complexity, with confidence of success.

7. Acknowledgments

This work was supported by grants from CNR-Progetto Strategico Reti Neurali, GNCB-CNR, Consorzio INFM and MURST. Part of this work was carried out in Summer 1993, while F. Masulli was a Visiting Scientist at the International Computer Science Institute at Berkeley (USA). We thank Ethem Alpaydin and Fabrizio Vannucci for helpful discussions.

8. References

1. E. Barnard, F. Kanaya, and S. Miyake. Comments on 'Bayes statistical behavior and valid generalization of pattern classifying neural networks' (with reply). *IEEE Transactions on Neural Networks*, 3:1026–7, 1992.

2. F. Casalino, F. Masulli, Sperduti, and F. A., Vannucci. Semantic phase transition in a classifier based on an adaptive fuzzy system. In *Proceedings of the Third IEEE International Conference on Fuzzy Systems, IEEE-FUZZ94*, Orlando, Florida, 1994, (in press).

3. G. Cybenko. Continuous valued neural networks with two hidden layers are sufficient. Technical report, Department of Computer Science, Tufts University, Medford, MA, 1988.

4. V. Di Gesu, R. De La Paz, W.A. Hanson, and R. Bernstein. Clustering algorithms for mri. In *Proceedings of Medical Informatics Europe 1991.*, pages 534–539, Vienna, Austria, 19-22 Aug. 1991, 1991. Springer-Verlag, Berlin, Germany.

5. R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, 1973.

6. J.A. Feldman and D.H. Ballard. Connectionist models and their properties. *Cognitive Science*, 6, 1982.

7. K. Fukunaga. *Introduction to statistical pattern recognition*. Academic Press, Boston, 1990.

8. M.D. Garris and R.A. Wilkinson. *NIST Special Database3 Handwritten Segmented Characters*. National Institute of Standard and Technology, Gaithesburg, MD , USA, 1992.

9. J. Hampshire and B. Pearlmutter. Equivalence proofs for multi-layer perceptron classifiers and the Bayesian discriminant function. In D.S. Touretzky, G. Hinton, and T. Sejnowski, editors, *Connectionist models : Proceedings of the 1990 Summer school*, pages 13–19, Denver, 1990. Morgan Kaufmann, San Mateo.

10. P.E. Hart. The condensed nearest neighbor rule. *IEEE Transactions on Information Theory*, 2:515–516, 1968.

11. J. Hertz, A. Krogh, and R.G. Palmer. *Introduction to the Theory of Neural Computation*. Addison Wesley, Redwood City, California, 1991.

12. J.J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences, USA*, 79, 1982.

13. J.J. Hopfield. Neurons with graded responses have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences, USA*, 81, 1984.

14. C.C. Jou. On the mapping capabilities of fuzzy inference systems. In *IJCNN International Joint Conference on Neural Networks*, pages 703–713, Baltimore, MD, USA, 7-11 June 1992, 1992. IEEE, New York, NY.

15. C.C. Jou. Comparing learning performance of neural networks and fuzzy systems. In *IEEE International Conference on Fuzzy Systems*, pages 1028–1033, San Francisco, 1993. IEEE, New York, NY.

16. F. Kanaya and S. Miyake. Bayes statistical behavior and valid generalization of pattern classifying neural networks. *IEEE Transactions on Neural Networks*, 2:471–475, 1991.

17. T. Kohonen. *Content-addressable memories*. Springer-Verlag, Berlin, 2 edition, 1987.

18. T. Kohonen. *Self-Organization and Associative Memory*. Springer-Verlag, Berlin, 3 edition, 1989.

19. B. Kosko. Fuzzy systems as universal approximators. In *IJCNN International Joint Conference on Fuzzy Systems*, pages 1553–1162, Baltimore, MD, USA, 7-11 June 1992, 1992. IEEE, New York, NY.

20. B. Kosko. *Neural networks and fuzzy systems : a dynamical systems approach to machine intelligence*. Englewood Ciffs Prentice Hall, NJ, 1992.

21. Y. Lee. Handwritten digit recognition using k nearest-neighbor, radial-basis function, and backpropagation neural networks. *Neural Computation*, 3:440–449, 1991.

22. R.P. Lippmann. An introduction to computing with neural nets. *IEEE ASSP Magazine*, pages 4–22, April 1987.

23. F. Masulli, F. Casalino, and F. Vannucci. Bayesian properties and performaces of adaptive fuzzy systems in pattern recognition problems. In *Proceedings of the European Conference on Artificial Neural Networks, ICANN-94*, Sorrento, Italy, 1994, (in press).

24. F. Masulli, F. Vannucci, and M. Penna. Learning with incremental input dimensionality in handwritten character recognition. In E. Lasker, editor, *Proceedings of the 4th International Symposium on System Research, Informatics and Cybernetics*, Baden-Baden, Germany, 1993, (in press). IIASS.

25. W.S. McCulloch and W. Pitts. A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5, 1943.

26. M.L. Minsky and S.A. Papert. *Perceptrons*. MIT Press, Cambridge, 1969.

27. S. Miyake and F. Kanaya. A neural network approach to a bayesian statistical decision problem. *IEEE Transactions on Neural Networks*, 2:538–540, 1991.

28. Nils J. Nilsson. *The mathematical foundations of learning machines*. Morgan Kaufmann, San Mateo, Calif., 1990.

29. Theo Pavlidis. *Algorithms for Graphics and Image Processing*. Springer-Verlag, 1982.

30. J. Platt. A resource-allocating network for function interpolation. *Neural Computation*, 3:213–225, 1991.

31. T. Poggio and F. Girosi. Regularization algorithms for learning that are equivalent to multilayer networks. *Science*, 247:978–982, 1990.

32. F. Rosenblatt. *Principles of Neurodynamics*. Spartan, New York, 1962.

33. D.W. Ruck, S.K. Rogers, M. Kabrisky, M.E. Oxley, and B.W. Suther. The multilayer perceptron as an approximation to a Bayes optimal discriminant function. *IEEE Transactions on Neural Networks*, 1:296–298, 1990.

34. D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning internal representations by error propagation. In D.E. Rumelhart and J.L. McClelland, editors, *Parallel Distributed Processing*, volume 1, chapter 8, pages 318–362. MIT Press, Cambridge, 1986.

35. D.E. Rumelhart, J.L. McClelland, and the PDP Research Group. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1. MIT Press, Cambridge, 1986.

36. R. J. Schalkoff. *Pattern recognition : statistical, structural, and neural approaches*. J. Wiley, New York, 1992.

37. C.W. Therrien. *Decision, Estimation and Classification*. John Wiley & Sons, New York, Chichester, Brisbane, Toronto, Singapore, 1989.

38. B. Widrow. Generalization and information storage in networks of adaline "neurons". In M.C. Yovits, G.T. Jacobi, and G.D. Goldstein, editors, *Self-Organizing Systems 1962*, pages 435–461, Chicago 1962, 1962. Spartan, Washington.

Appendix: Similarity Measures

In this appendix, we show some of the pattern similarity measures most widely used in the area of pattern recognition[17,36] :

- *Correlation.*

$$C(\mathbf{x}^a, \mathbf{x}^b) = \sum_{i=1}^{N} x_i^a x_i^b \ . \tag{35}$$

If $\mathbf{x}^a$ e $\mathbf{x}^b$ are two vectors of an Euclidean space, then C is the scalar product of such vectors.

$$C(\mathbf{x}^a, \mathbf{x}^b) = \mathbf{x}^a \cdot \mathbf{x}^b \ . \tag{36}$$

- *Direction Cosines.*

Let $\mathbf{x}^a$ and $\mathbf{x}^b$ be two vectors of an Euclidean space; their direction cosine is defined as:

$$cos\,\theta = \frac{\mathbf{x}^a \cdot \mathbf{x}^b}{\| \mathbf{x}^a \| \ \| \mathbf{x}^b \|} \ . \tag{37}$$

If the norms of vectors $\mathbf{x}^a$ and $\mathbf{x}^b$ are standardized to unity, then $cos\,\theta = C$.

- *Euclidean Distance.*

Let $\mathbf{x}^a$ and $\mathbf{x}^b$ be two vectors of an Euclidean space; their Euclidean distance is:

$$\| \mathbf{x}^a - \mathbf{x}^b \| = \sqrt{\sum_{i=1}^{N} (x_i^a x_i^b)^2} \ . \tag{38}$$

Note that

$$\| \mathbf{x}^a - \mathbf{x}^b \|^2 = \| \mathbf{x}^a \|^2 + \| \mathbf{x}^b \|^2 - 2\, \mathbf{x}^a \cdot \mathbf{x}^b \ . \tag{39}$$

Therefore, if the two vectors are standardized to unity:

$$\| \mathbf{x}^a - \mathbf{x}^b \|^2 = 2\,(1 - C) \ . \tag{40}$$

- *Generalized Hamming Distance.*

Let us consider two ordered sets of elements, with discrete or symbolic values; the generalized Hamming distances is the number of different symbols between the two sets.

For instance , consider the two ordered sets $\mathbf{x}^a$ and $\mathbf{x}^b$:

$$\begin{aligned} \mathbf{x}^a &= (p, a, a, s, t, q, b) \ , \\ \mathbf{x}^b &= (p, b, a, t, t, q, p) \ ; \end{aligned} \tag{41}$$

their generalized Hamming distance is then

$$H(\mathbf{x}^a, \mathbf{x}^b) = 3 \ . \tag{42}$$