Circular Backpropagation Networks for Classification

Sandro Ridella, *Member, IEEE*, Stefano Rovetta, and Rodolfo Zunino, *Member, IEEE*

***Abstract*— The class of mapping networks is a general family of tools to perform a wide variety of tasks; however, no unifying framework exists to describe their theoretical and practical properties. This paper presents a standardized, uniform representation for this class of networks, and introduces a simple modification of the multilayer perceptron with interesting practical properties, especially well suited to cope with pattern classification tasks. The proposed model unifies the two main representation paradigms found in the class of mapping networks for classification, namely, the *surface-based* and the *prototype-based* schemes, while retaining the advantage of being trainable by backpropagation. The enhancement in the representation properties and the generalization performance are assessed through results about the worst-case requirement in terms of hidden units and about the Vapnik–Chervonenkis dimension and Cover capacity. The theoretical properties of the network also suggest that the proposed modification to the multilayer perceptron is in many senses optimal. A number of experimental verifications also confirm theoretical results about the model's increased performances, as compared with the multilayer perceptron and the Gaussian radial basis functions network.**

***Index Terms*—Feedforward neural networks, backpropagation, pattern classification, knowledge representation**

I. Introduction

MAPPING neural networks [1] are computing devices that implement, in a distributed way, a function ψ, from some input domain $\mathcal{D} \subset \mathbb{R}^d$ to some output domain $\mathcal{T}$, parameterized by a set of parameters and featuring only feedforward signal paths. Such a general definition describes the basic properties common to many neural models, including virtually all networks used in practical applications. However, a unified theory of neural models does not yet exist.

A central topic in pattern recognition is classification. Mapping networks are widely used to approach classification problems when the task is to derive a rule from a set of examples. In classification tasks, the mapping to be learned represents a law that assigns a label to each pattern. Therefore, the output domain is defined as $\mathcal{T} = \{0,1\}^b$. In the following, by default a two-class problem will be assumed (hence $b = 1$); the general case with more than one output label will be considered only in statements involving a quantification of the number of parameters.

The problem of regression, that is, function approximation, has also been studied in great detail [2]–[4]. However, the present work addresses only the problem of classification, and focuses on the previously defined class of networks. We attempt to set up a framework to allow the study of a more general network model that may encompass different representation paradigms.

A layered mapping network can be described from a topological point of view by the number and dimensions of its layers and from a functional point of view by the transfer function of its units. The overall function ψ is then described in terms of a number of "simple" components. This description can be further detailed, and still encompasses a large number of neural-network models in use. A new scheme is proposed that on one side may help interpret a learned mapping, and, on the other hand, features interesting properties by itself.

Nonlinear discriminant functions have been considered by many authors. The circular unit model was introduced in the 1960's [5]. This and other works aimed at obtaining the best representation and memorization from single-layer networks. The related problem of generalization was introduced by Cover's paper. Vapnik [6] started a theoretical analysis of the topic, but a complete treatment was presented only recently [7]. A perspective similar to that of Cover's paper also characterizes the well-known book by Duda and Hart [8] (first edition), in which considerations about the degree-complexity of polynomial discriminant functions are presented. Geometrical learning procedures presented in [9] led to a method based on circular discriminant functions. A multilayer version was used in [10], but no theoretical analysis of the model is present. The circular unit was used in [11] for minimum-cost structures for classification, and in [12] for an interesting cascaded-architecture algorithm, whose complex topology prevents an easy interpretation of the classification rule synthesized.

The approach taken in many of these works is to consider polynomial activation functions as an alternative to the multilayer scheme. However, this introduces the need for additional constraints to keep the generality of the set of functions implementable by the model low enough for a good generalization [7], [13]. The present work aims instead to search for the *minimal* increment in the generality of the multilayer model that is capable of substantially improving the representation ability without affecting (and possibly enhancing) the generalization properties.

It should be stressed that the works focusing on radial basis functions [4], [14] are substantially different from the proposed approach, in spite of the formal equivalence. The present work includes a proof of the fact that the proposed model can be made equivalent to the Gaussian radial basis function network, thus demonstrating its generality. However, the radial basis function approach originates from the application of regularization techniques, whereas the present work focuses on

Manuscript received January 5, 1996; revised June 27, 1996. This work was supported by the Italian Ministry for the University and Research (MURST).

The authors are with the Department of Biophysical and Electronics Engineering, University of Genova, 16145 Genova, Italy.

Publisher Item Identifier S 1045-9227(97)00235-X.

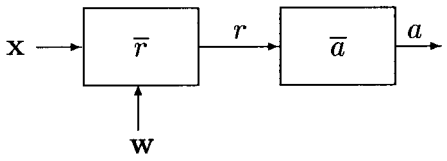

Fig. 1. A generalized model of the neural unit.

discriminant functions and their implementation by multilayer perceptrons. Moreover, a major distinguishing feature is the ability of the proposed model to implement a larger set of functions than that realizable by standard radial basis function approaches.

II. A Unified View of Mapping Networks

A. Generalized Neural Unit

In the present work we refer to the multilayer mapping network model with a topological structure inherited from the MLP. A single hidden layer will always be assumed in the following, without loss of generality. The network structure being fixed, we focus on the description and design of the (hidden) unit.

A generalized unit scheme is illustrated in Fig. 1, along with the symbols adopted. This scheme was introduced in [15]. $\boldsymbol{x}$ is the input vector of dimension d. The parameters (weights, bias, etc.) are the components of the vector $\boldsymbol{w} \in \mathbb{R}^p$, which needs not (and usually does not) have the same dimension as $\boldsymbol{x}$. The two blocks compute functions denoted by $\overline{r}$ and $\overline{a}$. The first block outputs the value $r = \overline{r}(\boldsymbol{x}, \boldsymbol{w})$, which we call the *stimulus*. The second block outputs the *activation* $a = \overline{a}(r)$.

These two quantities have a quite straightforward interpretation in geometric terms. The stimulus results from the application of a "filter" sensitive to some geometric property of the input space. The activation is the response of the unit to the geometric property pointed out by the stimulus.

Through the selection of appropriate functional forms for $\overline{r}$ and $\overline{a}$, the model can be used to represent all neural units usually adopted in practical applications. Some examples follow.

- The perceptron [16]: $r = \overline{r}(\boldsymbol{x}, \boldsymbol{w}) = w_0 + \Sigma_{i=1}^{d}\, x_i w_i; a = \overline{a}(r) = \mathcal{H}(r)$ (where $\mathcal{H}$ is a Heaviside function).
- The sigmoidal multilayer perceptron unit [17]: $r = \overline{r}(\boldsymbol{x}, \boldsymbol{w}) = w_0 + \Sigma_{i=1}^{d}\, x_i w_i; a = \overline{a}(r) = \sigma(r)$ (where σ is a sigmoidal function).
- The radial basis (Gaussian) unit [18]: $r = \overline{r}(\boldsymbol{x}, \boldsymbol{w}) = \|\boldsymbol{x} - \boldsymbol{c}\|^2/\sigma^2; a = \overline{a}(r) = e^{-r}$.

By using the terminology introduced, it is possible to make a parallel analysis of many network models by comparing their stimuli and activations. From the standpoint of function representation, however, a direct comparison of the mappings obtained by different networks is appropriate only in terms of global evaluation parameters, such as a properly defined functional distance, and not at the single unit level. A geometric or algebraic interpretation of the unit functions does not help in this respect. Hence an alternative interpretation of the underlying elementary components is required. The notion of *representation paradigm* can help obtain such an interpretation.

The representation paradigm is a characterization of mapping networks that is closely related to the geometrical properties of the stimulus. A distance-based stimulus (e.g., the Euclidean distance between the parameter vector and the input vector) can be associated with the *prototype-based* paradigm, according to which a network stores representative patterns (prototypes) and computes its output by measuring the match between a pattern and the stored prototypes. Nearest neighbor classifiers [19] implement this paradigm.

By contrast, the *surface-based* paradigm is represented by those models that draw region borders (hypersurfaces) in the input space, usually composed of individual segments realized by different units, and compute their output according to the position of an input pattern with respect to the borders. The perceptron [16] is an example of this paradigm.

The prototype-based paradigm can be regarded as being *data-oriented*, in that it represents data directly and decision boundaries only indirectly. The surface-based paradigm represents boundaries directly, and only indirectly data, so it is *rule-oriented*. The two approaches can be regarded as being complementary.

B. The Circular Unit and the CBP Network

The perceptron can be generalized by letting $\overline{r}(\boldsymbol{x}) = \Sigma_{i=1}^{p}\, w_i \xi_i = \boldsymbol{w} \cdot \boldsymbol{\xi}$, where the map $\boldsymbol{x} \mapsto \boldsymbol{\xi}$ ($\boldsymbol{\xi} \in \mathbb{R}^p$) is such that each component ξ_i is given by a product of components of $\boldsymbol{x}$, which can be some power of a single component or the product of powers of different components. Usually, one of the terms is a constant whose weight implements the bias. The parameter vector is of the same dimensions as $\boldsymbol{\xi}$, and the resulting stimulus is a polynomial with its components as coefficients.

This model is often adopted as a single-layer network scheme, as, for instance, in [5] and in more recent works, including the theoretical overview presented in [20]. In principle, its strength is the representation power, as every function may be at least locally approximated with arbitrary precision by a polynomial (e.g., Taylor's series expansion). However, it is not possible to avoid a very sharp increase in the number of terms required when the order is increased because it is not possible to select a priori some terms and to neglect the others (see [8]). For instance, the first-order model with bias (perceptron) has about as many parameters as inputs: $p = 1 + d$. For the complete second-order model, however, $p = 1 + d(d+1)/2$, which is of order d^2. In the general case, the number of terms of a complete polynomial with d variables of order q is $p = \binom{d+q-1}{q}$, which is of order d^q. The polynomial growth can be acceptable in cases with very small input dimension d, but it should be avoided for practical cases.

Other specialized unit functions can be devised based on requirements imposed by specific problems, the wavelet network [21] being an example. However, they often show a limited applicability for various reasons, e.g., some are too specific and tailored to a class of applications, and some others require a nonstandard training method.

On the basis of the above considerations, it is necessary to impose some constraints on the design of the stimulus and activation functions, if the model must be general and cost-effective at the same time.

- The overall network should have an increased representation power as compared with the standard MLP.
- The increase in the representation power should not affect significantly the generalization power; in other words, it should not cause an excessive increase in the probability of overfitting.
- The representation should allow for an easy interpretation of acquired knowledge. This is needed in order to use the network as a data analysis tool.
- The network should be trainable by a standard algorithm, without requiring a new theory.
- The implementation should stick as much as possible to the standard multilayer perceptron structure. This is especially important when dealing with hardware realizations, as we want to take advantage of the great efforts previously made by many researchers in the design of MLP hardware.

We consider the selection of an appropriate number of polynomial terms as the most sensible way to obtain a good compromise among the above constraints. In the following, the *circular backpropagation* (CBP) model [15] will be studied from this standpoint. As previously remarked, the model features the standard multilayer topology with a single hidden layer. At the unit level, the CBP model is described by the following functions:

$$\overline{r}(\boldsymbol{x}, \boldsymbol{w}) = w_0 + \sum_{i=1}^{d} x_i w_i + w_q \sum_{i=1}^{d} x_i^2 \tag{1}$$

or, expressing the quadratic term in the compact form $x_q = \Sigma_{i=1}^{d}\ x_i^2$

$$\overline{r}(\boldsymbol{x}, \boldsymbol{w}) = w_0 + \sum_{i=1}^{d} x_i w_i + w_q x_q \tag{2}$$

and

$$\overline{a}(r) = \sigma(r). \tag{3}$$

This is a special case of the polynomial unit described above. There is one additional parameter, i.e., the coefficient w_q, which weights the sum of the squared inputs. By simple algebraic transformations, it is possible to obtain another form for the same stimulus

$$r = g(||x - c||^2 - \theta) \tag{4}$$

in which the parameters $\boldsymbol{c}, g$, and θ do not appear as weights but have the following geometrical interpretations.

The distance from the point $\boldsymbol{c}$ in the space of inputs is computed and compared with the value θ. The result is scaled with the coefficient g to obtain the actual stimulus r, and the activation a is computed by the standard sigmoidal function. The output of the unit can be positive inside (for $g<0$) or outside (for $g>0$) a circular (in general, hyperspherical) region; anyway, a localized "bump" with a circular section is obtained around the point $\boldsymbol{c}$. Therefore, we describe the parameters as follows:

$\boldsymbol{c}$ = center or prototype

θ = radial threshold (hence $\rho = \sqrt{\theta}$ = radius)

g = gain.

We call these the "circular parameters." The transformation from standard perceptron weights to circular parameters is presented in Table I. The calculations involved are very simple, but for completeness they are presented in Appendix I.

TABLE I
RELATING WEIGHTS TO CIRCULAR PARAMETERS

From circular parameters to weights
$w_q = g$
$w_i = -g c_i$
$w_0 = g\left(\sum_{i=1}^{d} c_i^2 - \theta\right)$
From weights to circular parameters
$g = w_q$
$c_i = -w_i / 2w_q$
$\theta = \frac{1}{w_q}\left(\sum_{i=1}^{d} \frac{w_i^2}{4w_q} - w_0\right)$

The double form of each parameter is not a formal artifice, in that we adopt it to reflect the double nature of the representation. The circular parameters represent a transfer function implementing the prototype-based paradigm. However, when the coefficient w_q is very small, the circular parameters are not adequate anymore, and the stimulus collapses to the standard linear perceptron stimulus. In this situation, the unit implements the surface-based paradigm.

The choice between the two representation forms is dependent only on the value of adaptable parameters, so it is left to the optimization process. We refer to this fact by saying that the CBP model has a *paradigm plasticity* that enables it to adapt the representation form, without need for the user's supervision.

C. A Note on the Implementation of the CBP Network

By simple inspection, it is easy to see that the only different feature of a CBP network, as compared with the MLP, is

an additional input. This means that a CBP network can be obtained by an *off-line* modification to the training set, i.e., by adding the quadratic term x_q directly to the input patterns.

This trivial but fundamental observation allows one to devise very efficient implementations, in both hardware implementations and software simulations, if an MLP device is already available. The resulting network will be trained by plain backpropagation, at the only expense of an additional input (for a network with h hidden units, this means h additional weights).

However, when we are interested in issues related to the representation paradigm, and not in implementation details, we will consider the CBP model as being completely different from the MLP.

III. Properties of the CBP Model

This section describes the CBP model from different standpoints, including the results on the capacity, introduced by Cover [5], and on the Vapnik–Chervonenkis dimension [22], for both the single circular unit and the layered network. A very simple procedure to analyze a trained network to search for significant rules will also be presented.

As is well known, the Vapnik–Chervonenkis dimension (d_{VC}) of a learning machine is the maximum sample size such that there is at least one pattern set for which every dichotomy is implementable by the machine; the capacity $\mathcal{C}$ is the maximum sample size such that a pattern set for which every dichotomy is implementable has a probability 1/2.

There are many theoretical results that make use of d_{VC} as a sort of generalized number of degrees of freedom; for instance, a well-known result, although the estimated bounds are not very tight, is presented in [23]. Vapnik's learning theory [7] is a very sound and general background for classification and other learning problems; this accounts for the greater importance of d_{VC}, as compared with Cover's capacity. Notwithstanding these limitations, a number of results make use of the capacity $\mathcal{C}$, therefore, it could be exploited for comparisons with other models.

In the following, we shall prefix the name of a unit (for instance, perceptron) by using a symbol indicating the activation function: σ for sigmoidal activation, $\mathcal{H}$ for Heavisides. The same will be done for the names of multilayer networks (either CBP or MLP), for which the symbols will indicate the activation functions of the hidden units.

A. The Circular Unit

The capacity of a perceptron and that of a circular unit have been studied by Cover [5]. In the reference, it is shown that $\mathcal{C} = 2(d+1)$ for the perceptron and $\mathcal{C} = 2(d+2)$ for the circular unit.

It is known that, for a perceptron with inputs in $\mathbb{R}^d$, $d_{\mathrm{VC}} = d+1$ (a proof is given, for instance, in [24]). It is also known that, for d-dimensional hyperspheres, $d_{\mathrm{VC}} = d+2$ [25]. These results can be proved in many ways. In the following, we present a very simple and intuitive proof that requires only elementary linear algebraic considerations.

Theorem 1: A d-input linear threshold machine ($\mathcal{H}$-perceptron without bias) has $d_{\mathrm{VC}} = d$.

Proof: A linear threshold machine $\boldsymbol{w} \in \mathbb{R}^d$, when the vector $\boldsymbol{x}$ is fed at its input, outputs the value $\mathcal{H}(\boldsymbol{x}^T \cdot \boldsymbol{w})$. The machine is requested to learn the dichotomy T on the pattern set D of size n, defined by $T(\boldsymbol{x}) \in \{-1, +1\}\ \forall \boldsymbol{x} \in D$. The dichotomy T induces a vector $\boldsymbol{t}$ of size n, with binary components, such that $\mathrm{sign}(t_l) = T(\boldsymbol{x_l})\ \forall \boldsymbol{x_l} \in D$, and D can be arranged in a matrix X of d columns and n rows such that $\mathrm{row}_l(X) = \boldsymbol{x_l}$. There are 2^n possible dichotomies. The dichotomy T is implementable by the machine $\boldsymbol{w}$ if there exists an assignment for $\boldsymbol{w}$ such that

$$X\boldsymbol{w} = \boldsymbol{t}. \tag{5}$$

The value of d_{VC} is the maximum sample size such that there exists a pattern set D for which every T is implementable. For (5) to have a solution, the target vector $\boldsymbol{t}$ should belong to the column space of X (which is at most of dimension n). This is guaranteed for every $\boldsymbol{t}$, as long as the dimension of the space is greater than, or at least equal to, the number of columns ($n \leq d$). In this case, it is always possible to choose D such that X has full rank. Hence, $d_{\mathrm{VC}} \geq d$. Conversely, if $n > d$, the number of columns of X is insufficient for them to form a base in an n-dimensional space. Since the 2^n target vectors span the whole $\mathbb{R}^n$, then (5) cannot have a solution, whatever the choice of D. Hence $d_{\mathrm{VC}} = d$. ■

Proposition 1: Given a mapping ϕ: $X \rightarrow \Xi$, with $X \subset \mathbb{R}^d$ and $\Xi \subset \mathbb{R}^p$, if ϕ is nonlinear, then a linear $\mathcal{H}$-perceptron with input $\boldsymbol{\xi} = \phi(\boldsymbol{x})$ ($\boldsymbol{x} \in X, \boldsymbol{\xi} \in \Xi$) has $d_{\mathrm{VC}} = p$. If ϕ is linear ($\boldsymbol{\xi} = \phi\boldsymbol{x}$, where ϕ is a matrix of size (d, n)), then $d_{\mathrm{VC}} = \mathrm{rank}(\phi)$. In general, d_{VC} equals the number of linearly independent components of ϕ.

Proof: The proof follows directly from Theorem 1, by substituting $X' = \phi(D)$ for X. If ϕ is nonlinear, it is possible to choose D such that X' has full rank, whereas if it is linear, the maximum rank attainable by X' is $\mathrm{rank}(X\phi) \leq \mathrm{rank}(\phi)$. In the general case, it is possible to split ϕ into a linear part and a nonlinear part. The same considerations hold separately for the two parts. The result follows. ■

Corollary 1: A d-input affine threshold machine ($\mathcal{H}$-perceptron with bias) has $d_{\mathrm{VC}} = d+1$. A d-input $\mathcal{H}$-circular unit has $d_{\mathrm{VC}} = d+2$.

Proof: In an affine perceptron, $\phi(x) = w_0 + \Sigma_{i=1}^{d}\ w_i x_i$. There are $d+1$ terms, all linearly independent. In a circular unit, $\phi(x) = w_0 + \Sigma_{i=1}^{d}\ w_i x_i + w_q\ \Sigma_{i=1}^{d}\ x_i^2$. There are $d+2$ terms, all linearly independent. The direct application of Theorem 1 and Proposition 1 yields the result. ■

B. Representation Properties of the CBP Network

Since a CBP network features the MLP as a special case, it is possible to apply the known results on the approximation properties of the MLP to a CBP network in the case $w_q = 0$ (among others [3], [1]). Hence the general approximation properties of the MLP ensure that every mapping is realizable with a CBP net with arbitrary precision, since it is also realizable with an MLP. However, it is reasonable to expect the CBP model to have a higher representation power than that

of the MLP in terms of resources needed (number of hidden units, number of layers), by virtue of the properties shown at the unit level.

At this point, however, a distinction should be made between the cases of stepwise and continuous activations. In the multilayer case, it is known that the sigmoidal multilayer perceptron provides a representation power that is different from that of the hard-limited version (Heaviside activation function). Consequently, the estimate of the generalization power is also different. The case of sigmoidal activation will be briefly addressed later on.

Definition 1: Let S be a finite set, with elements in $\mathbb{R}^d$. Let $S^{(a)}$ be a subset of S and let $S^{(b)} = S - S^{(a)}$ be its complement with respect to S. Let H be a class of varieties (hypersurfaces) such that each element h induces a dichotomy in $\mathbb{R}^d$. The sets $S^{(a)}$ and $S^{(b)}$ are said to be *H-separable* if there exists $h \in H$ that realizes the dichotomy $S = S^{(a)} + S^{(b)}$.

Definition 2: (Special cases of separability) A dichotomy realizable by an element in the class $H = \{\boldsymbol{x} \in \mathbb{R}^d : \boldsymbol{x} \cdot \boldsymbol{w} = k\}$ of linear varieties (hyperplanes) is said to be *linearly separable*. A dichotomy realizable by an element in the class $H = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x} - \boldsymbol{c}\|^2 = k\}$ of isotropic second-order varieties (hyperspheres) is said to be *spherically separable*.

In the case of an $\mathcal{H}$-MLP, the result presented by Huang and Huang [26] holds:

Theorem 2 (Huang and Huang): Let S be a set of size $n < \infty$, with elements in $\mathbb{R}^d$. Let $\psi : S \rightarrow \{0, 1\}$ be an indicator function inducing an arbitrary dichotomy on S. There exists an $\mathcal{H}$-MLP network with $n - 1$ hidden units capable of realizing ψ.

A similar result can be stated for an $\mathcal{H}$-CBP network. However, the upper bound on the necessary number of hidden units is lower.

Proposition 2: If S is a finite set with elements in $\mathbb{R}^d$ every dichotomy, $\{\boldsymbol{x}\}, S - \{\boldsymbol{x}\}$ for $\boldsymbol{x} \in S$ is spherically separable.

Proof: From the hypothesis of finiteness of the set S, given $\boldsymbol{x}^* \in S$ we can state that there exists $r > 0$ such that, for each $\boldsymbol{x} \in S, \|\boldsymbol{x}^* - \boldsymbol{x}\| > r$. Hence it is always possible to construct a hypersphere h of radius r and center $\boldsymbol{x}^*$: $h = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x} - \boldsymbol{x}^*\|^2 = r\}$. ■

Proposition 3: Every linearly separable dichotomy is also spherically separable.

Proof: The proof is immediate if the linear separating hypersurface is written as a degenerate hypersphere of infinite radius and center to infinity. ■

This proposition is simply a geometric interpretation of the fact that the perceptron is a particular case of the CBP unit.

With the aid of these propositions, it is possible to state the following theorem, which is similar to Theorem 2 but refers to the case of circular units.

Theorem 3: Let S be a set of size $n < \infty$, with elements in $\mathbb{R}^d$. Let $\psi : S \rightarrow \{0, 1\}$ be an indicator function inducing an arbitrary dichotomy on S. There is a $\mathcal{H}$-CBP network with $\lfloor n/2 \rfloor$ hidden units capable of realizing ψ.

Proof: Let $S^{(0)} + S^{(1)}$ be the dichotomy on S induced by ψ: $S^{(0)} = \{\boldsymbol{x} \in S : \psi(\boldsymbol{x}) = 0\}, S^{(1)} = \{\boldsymbol{x} \in S : \psi(\boldsymbol{x}) = 1\}$. Consider singularly the elements of one of the subsets, say $S^{(1)}$, without loss of generality. According to Proposition 2, at most one hypersphere is required to separate each point $\boldsymbol{x} \in S^{(1)}$ from the whole $S^{(0)}$. It is also possible that a single hypersphere may separate two or more points; however, we are interested in an upper bound, so we search for the most unfavorable case. This is obtained when 1) n is even; 2) the sizes of $S^{(1)}$ and $S^{(0)}$ are both equal to $n/2$; and 3) no two points of $S^{(1)}$ are spherically separable from $S^{(0)}$. This requires $n/2$ hidden units. If n is odd, then it is possible to make this construction by using the subset of the smaller size, that is, $\lfloor n/2 \rfloor$. Hence the number of hidden units required is at most $\lfloor n/2 \rfloor$.

On the basis of this result, we can expect that the representation performance in "easy" cases will be similar to that of an MLP. But when the complexity of the rule to be learned increases, the number of hidden units for the CBP network will have to increase more slowly than for the MLP. When the worst case is reached, the number for the CBP network will be approximately half that for the MLP.

C. Extension to the Case of Continuous-Activation Units

The following result states that the representation properties of the MLP are enhanced when adopting the σ-version.

Theorem 4 (Sontag [27], [28]): Any dichotomy on a set of patterns of cardinality $2h$ can be implemented by some σ-MLP with h hidden units.

By this result, the gain of a $\mathcal{H}$-CBP network with respect to the $\mathcal{H}$-MLP configuration is comparable to that of switching from $\mathcal{H}$-MLP to σ-MLP. It is possible to find instances of a further improvement by considering σ-CBP networks. In these cases, the number of hidden units for a given number of patterns is even less than half.

An example of this kind is the "alternate labels" problem, arising naturally in the context of Sontag's work. This problem consists of a given number of data points, all lying on the same line. There are two possible class labels. Each point is labeled differently from its neighbors, so that the targets alternate along the line. We will consider the case of equispaced points throughout this work.

In the "alternate labels" case, for instance with 11 data points, it is experimentally demonstrated that a CBP network with $h = 3$ can solve the problem, while the above theorem indicates that a MLP needs $h = 6$. However, there are other circumstances in which the gain of CBP is not so large. We refer to the following recent result.

Theorem 5 (Sontag [29]): A network (either σ-MLP or σ-CBP) with p parameters can shatter any set of $2p + 1$ points in general position.

In this case, since only the number of parameters is taken into account, the user who wants to improve the representation performance of an MLP architecture can either increase the number of hidden units or step to the CBP model.

This discussion can be closed with a note on the practical side. While the "alternate labels" case can be mapped on the framework studied by Mirchandani [30], if each data point is made to represent the "center" of an input region, the "general position" is in fact very peculiar. Therefore, in applicative cases we may expect an intermediate situation, in which the

gain of σ-CBP with respect to $\mathcal{H}$-MLP is not as large as in the alternate labels example, but is larger than that of a simple σ-MLP.

D. Generalization in CBP Networks

A fundamental advantage of the CBP model is that most of the results known for the MLP can be stated for the CBP model, too, with appropriate modifications. This means that the fundamental estimate of d_{VC} for a multilayer $\mathcal{H}$-network, provided in [23], is still valid: for an MLP with d inputs, h hidden units, and b outputs, $d_{\mathrm{VC}} \leq 2(h(d+1)+b(h+1))\log(e(h+b))$. Hence, for a CBP net with the same topology, $d_{\mathrm{VC}} \leq 2(h(d+2)+b(h+1))\log(e(h+b))$. These values are not very different, and their ratio approaches one for networks with a large number of inputs, so the expected generalization ability is similar for the MLP and the CBP model.

The same reasoning is not so easy when one tries to estimate the capacity $\mathcal{C}$. One reason is that the definition of capacity cannot be generalized in a unique way [31]. However, some results are presented in [32] (a more detailed version is to appear [33]), and can be extended to the circular case by means of the same arguments that hold about the d_{VC}. In particular, for networks with one output unit, it is shown that $dh+1 \leq \mathcal{C} \leq 2(dh+1)$ in the case of the $\mathcal{H}$-MLP. These bounds are obtained for sets of points in general position. The transformation $\boldsymbol{x} \mapsto \boldsymbol{\xi}$ converting MLP into CBP (i.e., the addition of the sum-of-squares component to the input vector) does not always preserve the general position; in other words, if every subset of size d of the set of vectors $\{\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(n)}\}$ is linearly independent (does not lie on any d-dimensional hyperplane), this is not always true in the corresponding set $\{\boldsymbol{\xi}^{(1)}, \cdots, \boldsymbol{\xi}^{(n)}\}$. Therefore we can state the following bounds for the $\mathcal{H}$-CBP network: $dh+1 \leq \mathcal{C} \leq 2((d+1)h+1)$.

The lower bound could be tightened to $(d+1)h+1$ if we restricted the general position requirement to exclude also sets of points lying on d-dimensional hyperspheres; however, this is not very interesting, since the corresponding gain is proportional to h, but generally one tries to keep h as small as possible.

E. Knowledge Reversal by Interpretation of the Trained Network

In some cases, the internal representation of the MLP is used to extract informations about the structure of the classification mapping. This *rule extraction* is based on the interpretation of the transition between decision regions, at the unit level, as *if-then* rules.

The same reasoning can be applied to a CBP network. However, in this case, additional informations can be extracted from the representation paradigm adopted by each unit. In other words, if the weight w_q is very small, the unit is implementing a standard perceptron rule. If the weight w_q is not negligible, the unit is implementing a circular (distance-based) rule. Since the paradigm is decided by the optimization process, this gives an information on what type of representation better fits the training data.

This observation does not imply a reliable knowledge-reversal process in every situation; nonetheless, the rule-extraction procedures applicable in the standard MLP case are preserved by the CBP model, whereas the latter introduces the additional information on whether the rules are of the global or of the local type.

F. Optimality of the CBP Model

With reference to the list of desirable properties presented in Section II, this section has shown the following.

- The representation power of CBP is larger than that of the standard MLP.
- The increase in the representation power does not affect significantly the generalization power. The d_{VC} value estimated for CBP is very close to that estimated for the MLP.
- The representation allows for an interpretation of acquired knowledge in terms of representation paradigms. Therefore, the knowledge reversal is analogous, in the worst case, to that of the MLP, but is often easier.
- The network is trainable by standard backpropagation in a transparent way.
- The structure of the network is almost identical to that of the MLP.

The CBP model is a very special case in the class of polynomial units as described earlier in this section. In general, the ability to implement a localized activation allows the feasibility of a prototype-based representation. The polynomial model features this ability, but the order q must equal two. In the class of second-order units it is possible to have or not to have a localized activation; this depends on the relationships among coefficients, and may be verified by analyzing the definiteness of the matrix that describes the overall transfer function as a quadratic form.

To obtain the required ability, it is necessary to impose constraints on the coefficients of the second-order terms (powers of inputs and products of inputs). It is difficult to ensure that these constraints will be met in every case because they involve more than one weight. The only situation that guarantees the function to be able to implement localized activations is the circular one, as it involves one coefficient w_q for all the second-order terms.

Therefore, the CBP model is the only one that is capable of switching from a linear to a localized (circular) activation region without requiring additional control structures that constrain the parameters to satisfy special conditions. At the same time, it is also the least costly in terms of number of additional parameters, as compared with the MLP unit. Only in the circular case is the number of parameters p constant with d, $p = 2+d$. In the restricted elliptical case, for instance, it is linear in d: $p = 1+2d$. In the most general second-order polynomial case, p is quadratic in d, $p = 1+d+d^2/2$. It should be noted that the number of parameters should be kept as small as possible for many reasons, including those regarding learning time, storage cost, sample complexity/generalization power, and readability of the learned mapping.

To sum up, the CBP model is optimal in terms of *gain in representation power* (according to our requirements) versus *increase in the number of parameters*.

IV. Equivalence to Gaussian Radial Basis Function Networks

In this section we shall show that the CBP model may be made equivalent to another widely used neural scheme, i.e., the network of locally tuned Gaussian units.

Equivalence between two network models requires two conditions to be satisfied. The first is that the sets of functions implementable by the two models coincide. The second is that the training procedures should allow them to learn the same mapping for the same training set.

The first condition is of architectural nature. It can be verified by comparing the structure and interconnections of the layers, and the activation functions of the units. The second condition is related to the algorithms used for training and not to the networks. It can be verified by comparing the iterative learning steps. However, if the performance criterion adopted in training is the same for both models (e.g., in classification, the percentage of correctly labeled patterns), we can concentrate on the architectural equivalence, since the goal of the optimization process coincides in the two cases.

The transfer function of a circular unit is radially symmetric. Hence a CBP net has by itself the structure of an RBF network. However, in practice, the most commonly adopted basis functions are the isotropic Gaussians

$$\mathcal{G}(\boldsymbol{x}) = \exp\left(-\frac{||\boldsymbol{x}-\boldsymbol{c}||^2}{\sigma^2}\right). \tag{6}$$

The training of such networks requires the choice of appropriate values for the parameters $\boldsymbol{c}$ and σ, which is usually made independently. Here we show that a σ-CBP network can implement a Gaussian RBF network; therefore, backpropagation training can be used to obtain the same results as those obtained by RBF training.

Proposition 4: There is a two-layer σ-CBP network equivalent to a Gaussian RBF network with the same number of hidden units h.

Proof: The stimulus of the generic hidden unit of a σ-CBP network can be expressed in terms of the circular parameters as per (4). The activation function is

$$\overline{a}(r) = \frac{1}{1+e^{-r}}. \tag{7}$$

Therefore, if we let $r' = g||\boldsymbol{x}-\boldsymbol{c}||^2$, the overall transfer function of the unit can be expressed as

$$a = \frac{1}{1+e^{-(r'-g\theta)}}. \tag{8}$$

By some algebraic manipulations, this expression can be transformed as follows:

$$a = \frac{1}{1+e^{-r'}e^{-g\theta}} = \frac{e^{r'}e^{g\theta}}{e^{r'}e^{g\theta}+1.}$$

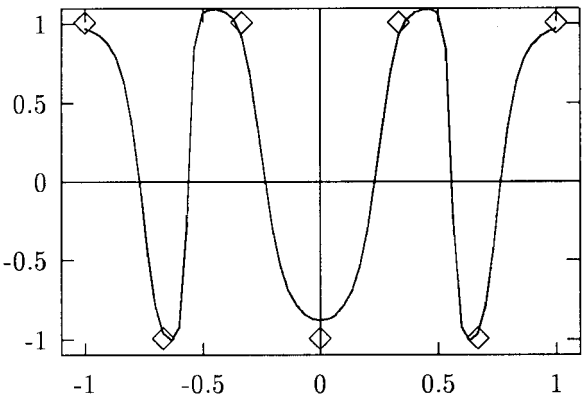

Fig. 2. How CBP solves the alternate-labels problem.

A generic output unit will not receive this value directly as an input, but only after a multiplication by the weight w. Therefore, the output value of the hidden unit can be multiplied by an arbitrary constant, which will be compensated for by the subsequent weight

$$ka = e^{r'}\frac{ke^{g\theta}}{e^{r'}e^{g\theta}+1.}$$

Let the term $g\theta$ take on very large values. Let the constant k take on correspondingly small values. The multiplying fraction can then take on values arbitrarily close to one. Hence, including the weight in the expression for the output value, we can write

$$|wa_{\mathrm{RBF}} - w'a_{\mathrm{CBP}}| < \epsilon$$

for any $\epsilon > 0$, where a_{RBF} is the activation computed by using the Gaussian activation function and stimulus as per (6), a_{CBP} is the activation using the CBP activation function and stimulus, w is the output weight, and w' is the compensated output weight $kw' = w$. ■

After showing that a CBP network can encompass also the Gaussian RBF model, we may ask whether the converse is also true, which means that the two approaches are theoretically identical. However, this is not the case. This may be shown with the aid of the alternate-labels problem. Fig. 2 shows an alternate-labels problem with seven data points, and the one-dimensional (1-D) activation profile of two CBP units.

It is possible to see that the CBP activation profile can identify seven zones, characterized by sign inversion, while RBF is limited to five zones. This has been shown theoretically for RBF (the proof is in Appendix II), and experimentally demonstrated for CBP, as shown in the figure, with good convergence rate.

We conclude with a note on the representation properties of the CBP activation function as compared with the Gaussian function. In the CBP network the parameters are expressed in the form of weights, rather than in the circular form. This means that degenerate radial functions are implementable in the CBP formalism, since an infinite radius is realizable when expressed as $w_q = 0$. In the RBF formalism, this would mean giving an infinite value to an actual parameter (the center's coordinates), which is unrealizable both in physical hardware

and in software simulation. This means that the equivalence between RBF and MLP could be theoretically assessed in the limit, but not physically attained, whereas the equivalence between CBP and MLP is feasible also in practice. Comparing the d_{VC} of CBP with that of RBF is difficult, since to the best of our knowledge no information on this topic is available for RBF networks in the classification framework. However, one can expect that the d_{VC} of RBF will be proportional to the number of weights in the network [34]. We stress that Theorem 5 remains valid also for the RBF activation function, if the training set is in general position.

V. Experimental Results

A. Experimental Setup

The simulations are grouped into three different sets. The first set of two tests aims at obtaining very simple verifications of the properties described theoretically. The second set is a more comprehensive study of several properties of CBP networks, based on a family of data sets generated by Gaussian mixture distributions. These experiments follow the approach presented by de Villiers and Barnard in [35] to allow a comparison with their results, obtained for the MLP. The third set is a standard benchmark, i.e., a vowel recognition task, available on-line in the repository of Carnegie Mellon University, Pittsgurgh, PA.[1] Although experimental comparison among different classification procedures is probably an ill-posed problem [36], our choice is to complement theoretical analysis with practical verifications. This allows a more complete description of the model under study.

The first two problems consist of two-dimensional (2-D) synthetic tests (for ease of visualization). The training sets are shown in Figs. 3 and 4. The first problem, a ten-points version of the "alternate labels" problem, aims at comparing the representation properties in the worst-case addressed in Theorem 3 for the MLP and CBP. The second problem is the well-known "two spirals" benchmark [37], [38], commonly adopted as a testbed for pattern classification systems. The data set consists of points belonging to two interspersed spiral-shaped classes, with 97 samples for each class.

The Gaussian mixtures are used to create a set of experiments, originally aimed at doing statistical considerations on the representation and generalization properties of MLP networks with different layouts (one- and two-hidden layer networks). Here we adopt the same approach in order to compare the CBP and MLP models. The training sets are random samples of mixture distributions, resulting from superpositions of equiprobable Gaussian clusters. The parameters of the Gaussian clusters (averages and variances) are in turn randomly selected from a Gaussian distribution. Patterns are d-dimensional, with $d \in \{2, 5\}$, and the distribution of each cluster is the product of d univariate Gaussian distributions; this means that the principal directions coincide with the coordinate axes. An example is given in Fig. 5. We refer the reader to [35] for a complete presentation of this "distribution of distributions."

[1] Anonymous ftp: ftp.cs.cmu.edu, directory /afs/cs/project/connect/bench.

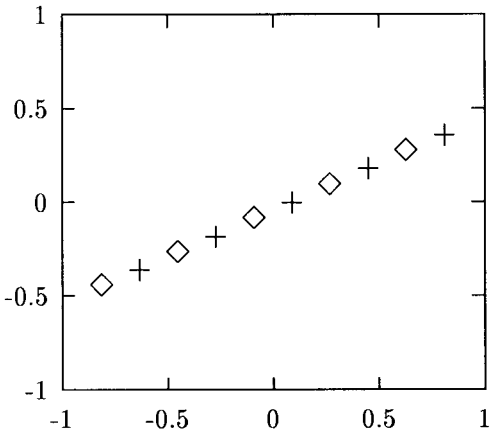

Fig. 3. The alternate-labels problem with ten data points.

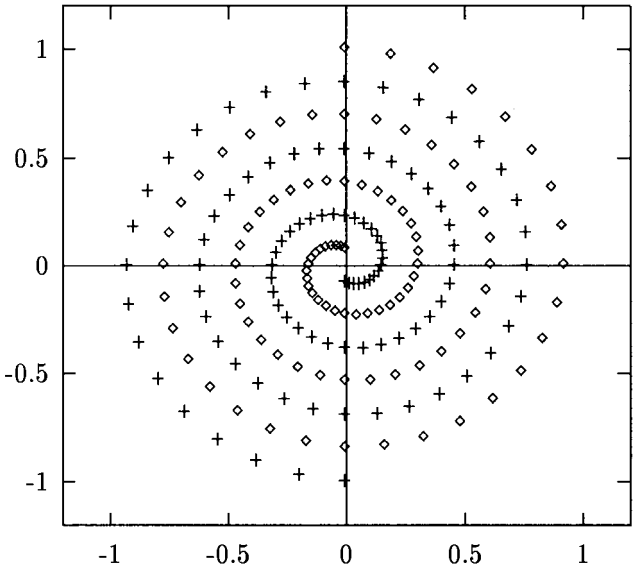

Fig. 4. The two-spiral problem.

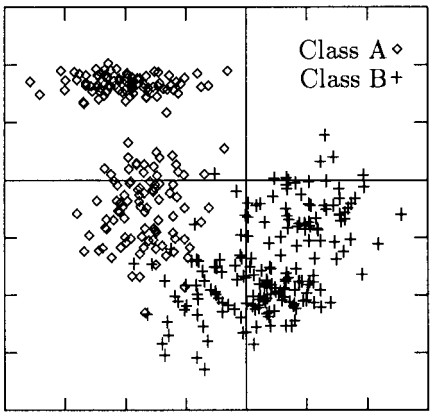

Fig. 5. A training set drawn from the Gaussian mixture distribution.

The vowel recognition task is based on the real-world data collected by Deterding [39] for speech recognition experiments. The data represent a ten-dimensional encoding of the steady-state part of vowels uttered by different speakers. There are 11 classes, corresponding to as many vowel sounds. The

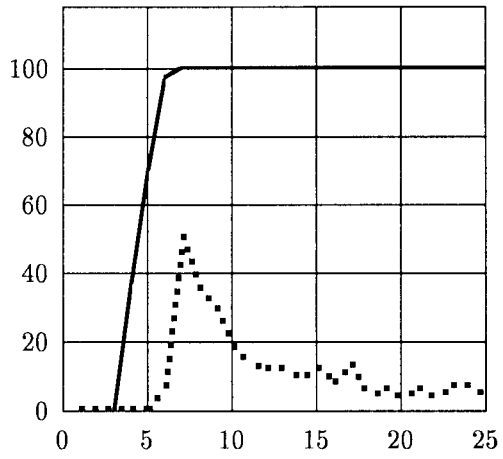

Fig. 6. Training results for the alternate-labels problem with ten data points. Percentage of convergence of multistart training for CBP (solid line) and MLP (dotted line) versus number of hidden units h.

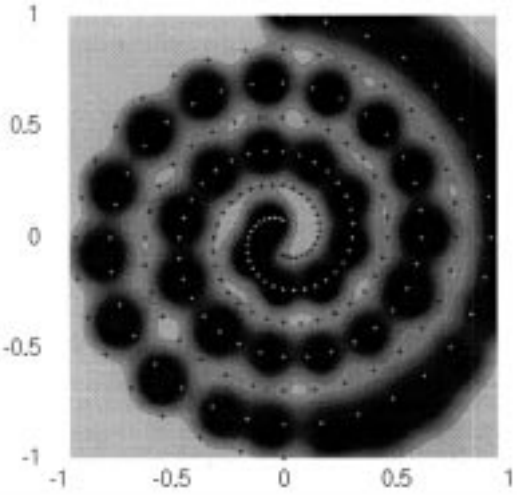

Fig. 7. Training results for RBF on the two spirals problem.

standard "vowel" database is composed of a training set (528 patterns) and a test set (462 patterns), to allow generalization estimation.

The backpropagation procedure adopted was accelerated by the method by Vogl *et al.* [40] for adapting the training parameters. An implementation of the algorithm, with optimizations for RISC architectures, is available online.[2]

The RBF tests have been performed with a network featuring a hidden unit activation of the form given in (6) rather than that of (4). A CBP network and a RBF network differ essentially in that the term θ in (4) is null in (6), and in that the term g in (4) is substituted for by $-1/\sigma^2$ in (6), that is necessarily of negative sign.

Training of this RBF structure is accomplished by gradient descent, as described above, with the derivatives of the cost function with respect to the parameters given by Bishop [41, pp. 190–191].

When a random variable was required, the random number generator presented in [42] was used. The Gaussian generator routine can be found in [43].

B. Results on the Two-Dimensional Problems

The alternate-labels problem turns out to be very difficult for standard backpropagation to learn. In the diagram of Fig. 6, a number of experiments with varying numbers of sigmoidal hidden units (h) are presented. Sigmoidal activations were chosen to allow backpropagation training. For each value of h, 1000 training trials were run, starting from different seeds. The training was stopped either at convergence or when the number of epochs reached the threshold of 20 000. The percentage of successful trials is plotted versus the value of h. We observe that, as expected (Section III), the presence of sigmoidal activation functions increments the capacity of the network, as compared with Heaviside units. This fact holds for both cases (MLP and CBP).

It is possible to see that the MLP does not converge 100% of the times for any value of h. The convergence rate corresponding to the minimum theoretical number of hidden units (i.e., $h = 3$ for CBP and $h = 5$ for MLP) is under 1%, therefore in the plot it is not possible to appreciate it. The actual percentages are .5% for CBP and .3% for MLP. The decrease in the plot can be ascribed to the fact that, when h increases, so does the number of parameters, therefore either the threshold of 20 000 epochs or the number of starts per training run should have been increased to cope with the more complex optimization problem. On the other hand, as soon as the theoretical requisites for the representation of configurations are met (i.e., h sufficient for the n points), CBP converges with little or no difficulty.

TABLE II
RESULTS FOR RBF ON THE TEN-POINTS ALTERNATE-LABELS PROBLEM

h	Min. error	Convergence
3	20%	0%
4	10%	0%
5	0%	4.4%

It is interesting to investigate the convergence of a RBF network on the same problem, to compare it with an equivalent CBP network. The results of this experiment are summarized in Table II. For each number of hidden units, the minimum error obtained in training (second column) and the percentage of zero-error trials (third column) are presented. The lower representation capacity of RBF with respect to CBP can explain the fact that RBF does not converge for $h<5$, as discussed in the previous section. For $h = 5$, performance of RBF equals that of MLP, in agreement with Theorem 4. Of course, in these conditions ($h = 5$ with $n = 10$), RBF training can be implemented in a much faster way by using a cluster analysis of the training set before starting the classification phase, instead of a plain gradient descent [14].

[2] ftp://risc6000.dibe.unige.it/pub; files mbp*.

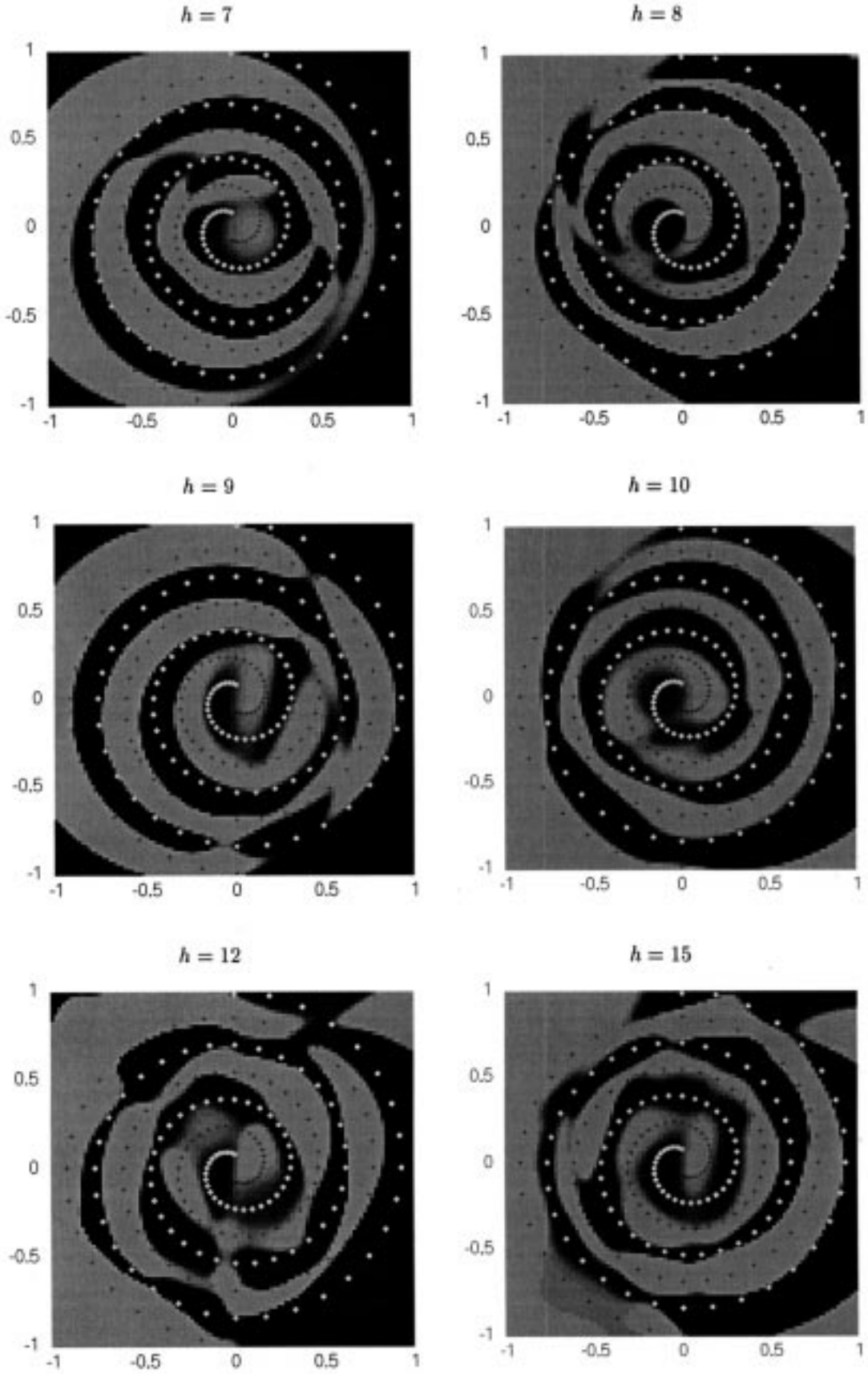

Fig. 8. Training results for the two spiral problem. Visualization of the output with varying number of hidden units h.

The performance of the MLP on the "two spirals" problem was reported in [37] for a network with three hidden layers of five units each. Here we show the results of training CBP networks with one hidden layer of seven to 15 units (Fig. 8).

TABLE III
NEIGHBORS IN THE "SPIRALS" DATA SET

HN	P	HN	P
0	21	6	2
1	0	7	4
2	44	8	2
3	2	9	2
4	12	10	5
5	3	Total 97	

All the trials were ended at convergence; therefore, seven hidden units in one layer are sufficient to solve this problem by the CBP model.

An RBF network has been trained on the same problem. The result is shown in Fig. 8 for $h = 42$. However, no convergence has been obtained with $h \leq 41$. The case of $h = 42$ requires considerable optimization efforts. We consider $h = 41$ as a threshold value, based on the following considerations.

We ask how many neighbors of the same class can be represented by each hidden unit. For each point of one of the two spirals we take into account its neighbors, starting from the nearest and proceeding according to their distance ranking. We count how many neighbors belong to the same spiral ("homogeneous" neighbors), before finding a point lying on the other spiral. The results are summarized in Table III, where HN indicates the number of homogeneous neighbors and P the number of points.

Using these data, the RBF network size can be estimated based on the fact that:

1) 20 hidden units represent isolated points;
2) 15 hidden units represent points with $HN = 2$;
3) six hidden units represent points with $HN \geq 3$.

Therefore, $h = 41$ is the minimum size for which a cluster analysis based on nearest neighbor consideration is practically feasible. This does not mean that smaller size nets could not be used; nevertheless, we can expect that the convergence rate will experience a steep decrease with decreasing h, since initialization becomes nontrivial under that threshold.

C. Results on the Gaussian Mixtures

The experiments were based on multistart training (ten trials per training run with different initializations). The measurements were obtained by averaging over multiple training set distributions (differing in both the number of clusters and their parameters), multiple samplings from each distribution, and multiple sample sizes (either 100 or 1000). The results are parameterized by the topology, hence they are a function of h. As in the original experiments, the number of weights was left constant for $d = 2$ and $d = 5$, and set to about 20, about 40, and about 60 (within 5% tolerance).

TABLE IV
RESULTS OF THE GAUSSIAN CLUSTERS EXPERIMENTS

	p	MLP	CBP	RBF
T_{ave}	20	65.22 (8.94)	96.28 (3.36)	94.93 (3.36)
	40	60.70 (5.82)	96.84 (3.17)	93.99 (4.96)
	60	63.02 (5.25)	96.58 (3.42)	93.69 (4.86)
G_{ave}	20	65.55 (13.99)	72.78 (12.98)	75.29 (7.60)
	40	67.68 (12.88)	73.79 (13.61)	81.09 (10.70)
	60	67.89 (12.41)	73.84 (14.34)	78.68 (10.84)
T_{opt}	20	69.29 (10.35)	97.58 (3.22)	96.23 (3.34)
	40	63.36 (6.43)	97.87 (2.92)	94.30 (4.75)
	60	65.20 (5.32)	97.70 (3.06)	94.24 (4.80)
G_{opt}	20	65.17 (16.03)	71.60 (14.70)	74.47 (11.77)
	40	66.87 (13.65)	73.54 (13.75)	79.87 (13.43)
	60	67.36 (12.33)	73.99 (14.26)	78.90 (12.05)

The parameters measured in these experiments are related to classification performances (percentage of correctly labeled patterns) over the 10 trials of each run, and are defined as follows (see also [35]): T_{ave} is the average training performance, T_{opt} is the best training performance;, G_{ave} is the average test performance, and G_{opt} is the test performance of the net that featured T_{opt}.

Table IV contains the estimated values of the parameters under study, with experimental standard deviation annotated in parentheses. These data are summarized in Fig. 9.

The results suggest that, *in this case*, the classification performance of a CBP network is always higher than that of an MLP network with the same number of hidden units. This holds true even on the test set, although it is commonly acknowledged that a model with a larger number of parameters is more subject to overfitting than a model with fewer parameters. We recall that for a three-layer MLP the number of weights is $p_{\text{MLP}} = (d+1)h+(h+1)b$, whereas for a CBP with the same topology it is $p_{\text{CBP}} = (d+2)h+(h+1)b = p_{\text{MLP}}+h$.

In the case of RBF, we can observe that performance on the test set is better than that of CBP. This could be explained by the fact that data are clustered with a Gaussian distribution, which could make it easier for the RBF networks to represent them (although the data are not necessarily isotropic). However, training results are slightly better for CBP.

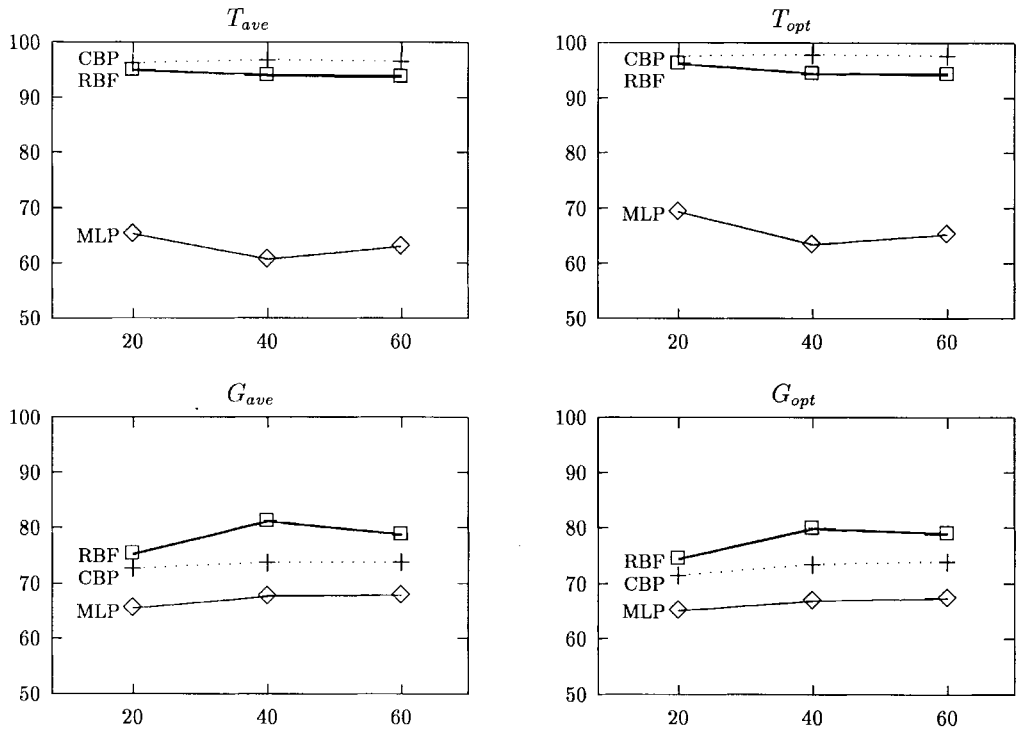

Fig. 9. Average results of training on Gaussian mixtures versus number of weights.

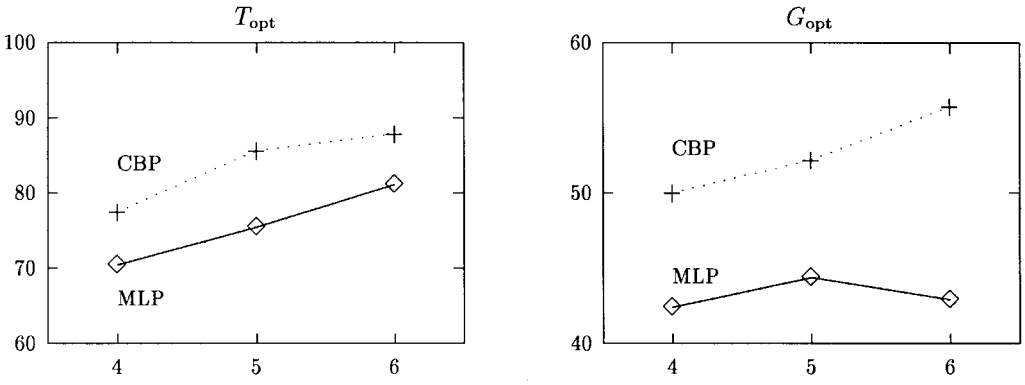

Fig. 10. Results of training on the "vowel" problem versus number of hidden units.

D. Results on Vowel Recognition

The results summarized in Fig. 10 and detailed in Table V were generated by a set of training runs. Several trials were performed. To facilitate the repeatability of the experiments, the results were obtained as follows: first, the minimum MSE was searched for; then, the corresponding classification error was recorded on both the training set (T_{opt}) and the test set (G_{opt}). This procedure is quite different from stopped training with cross-validation, since the test performance is not taken into account in the stopping criterion. The table compares the test error obtained by MLP and by RBF with that obtained by CBP. For this real-world benchmark, the CBP model learns substantially better than the MLP. This can be seen by comparing the approximation error and the classification error on the training set. The generalization ability of CBP (as estimated by this particular test) is also greater than that of the MLP. Results for RBF are in some way intermediate between those of MLP and of CBP.

TABLE V
RESULTS OF THE "VOWEL" DATABASE EXPERIMENTS

	MLP			CBP			RBF		
h	MSE	T_{opt}	G_{opt}	MSE	T_{opt}	G_{opt}	MSE	T_{opt}	G_{opt}
4	0.1452	70.4	42.4	0.1300	77.5	50.0	0.1127	77.5	44.4
5	0.1204	75.4	44.4	0.1084	85.6	52.2	0.0956	82.6	45.0
6	0.1045	81.1	42.9	0.0856	87.9	55.8	0.0756	85.0	33.8

The test performance, as compared with other results, seems unsatisfactory. However, it should be considered that the networks adopted did not feature more than six hidden units. In Table V results for networks with comparable numbers of hidden units are presented. The usual RBF approaches often involve larger networks. An example is presented in [44], where a very good performance (65% correct) is reported for

an RBF-type network with 204 units. To make a comparison, a CBP network with 80 hidden units was trained with stopped training by cross-validation, reaching the same value (65.1% correct).

VI. Concluding Remarks

In this paper, the properties of the circular backpropagation multilayer network have been investigated from the standpoint of pattern classification. Theoretical analysis and experimental evidence suggest that this model is especially well suited to implement classification tasks. The paradigm plasticity featured by the model allows the implementation of classification principles which have different interpretations, based either on the classification rules (by direct implementation of the decision boundaries) or on the data (by implementation of prototypes of the nearest neighbor type). Results about the properties of the model have been illustrated with experimental verifications, on both synthetic problems and a real-world benchmark.

The perspectives of research include a hardware implementation of the model which will be applied to a character recognition task. Hardware implementation is very simple, since it reduces to a preprocessing phase to be applied to the input of a standard multilayer perceptron network. The theoretical analysis is being extended to encompass other neural models (e.g., vector-quantization networks) within the same framework. This requires only simple modifications to the standard scheme, such as weight linking, so that the resulting networks are still trainable by plain backpropagation.

Appendix I
Transformation of Parameters

$$\begin{aligned}\overline{r}(\boldsymbol{x},\boldsymbol{w}) &= w_0 + \sum_{i=1}^{d} x_i w_i + w_q x_q \\ &= w_q\left(\frac{w_0}{w_q} + \sum_{i=1}^{d} x_i w_i + x_q\right) \\ &= w_q\left(\sum_{i=1}^{d}\left(\frac{w_i^2}{2w_q}\right)^2 + \sum_{i=1}^{d} x_i \frac{w_i}{w_q} + x_q\right) \\ &\quad + w_q\left(\frac{w_0}{w_q} - \sum_{i=1}^{d}\left(\frac{w_i^2}{2w_q}\right)^2\right) \\ &= g(||\boldsymbol{x}-\boldsymbol{c}||^2 - \theta)\end{aligned}$$

by defining the circular parameters as in Table I, and recalling that $x_q = \Sigma_{i=1}^{d}\ x_i^2$.

Appendix II
RBF Cannot Solve the Seven-Points Alternate-Labels Problem

In this Appendix we give a proof of the fact that the 1-D alternate-labels problem with seven equispaced data points cannot be solved with Gaussian RBF with $h = 2$.

Consider a Gaussian RBF network with $d = 1, h = 2, b = 1$ to attempt representing the alternate labels problem with seven data points. Symmetry considerations allow the stimulus of its output unit to be expressed as

$$r_{\text{out}} = w_0 + w_1 e^{-g_1 x^2} + w_2 e^{-g_2 x^2}. \tag{9}$$

Derivation of this expression with respect to x yields

$$\frac{\partial r_{\text{out}}}{\partial x} = -g_1 x w_1 e^{-g_1 x^2} - g_2 x w_2 e^{-g_2 x^2}. \tag{10}$$

This expression vanishes for $x = 0$, for $x = \pm\infty$, and for

$$x = \pm\sqrt{\frac{\ln(-w_1 g_1 / w_2 g_2)}{(g_2 - g_1)}}.$$

(This pair of roots is defined only when the arguments of the logarithm and of the root are both positive. We assume this is the case, since we are interested in assessing the maximum number of roots.)

The roots of the derivative correspond to minimum, maximum and saddle points. Between pairs of these points, we can identify at most five regions corresponding to five different classification outputs. Therefore the seven-points problem cannot be solved.

Acknowledgment

The authors gratefully acknowledge the anonymous reviewer's comments for stimulating the theoretical and experimental comparison between CBP and RBF.

References

[1] R. Hecht-Nielsen, *Neurocomputing*. Reading, MA: Addison-Wesley, 1989.
[2] K. Hornik, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, 1989.
[3] G. Cybenko, "Approximation by superposition of a sigmoidal function," *Math. Contr., Signals, Syst.*, vol. 2, pp. 303–314, 1989.
[4] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proc. IEEE*, vol. 78, pp. 1481–1497, Sept. 1990.
[5] T. Cover, "Geometrical and statistical properties of inequalities with application in pattern recognition," *IEEE Trans. Electron. Comput.*, vol. 14, pp. 326–334, 1965.
[6] V. N. Vapnik and A. J. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," *Dokl. Akad. Nauk SSR*, vol. 4, no. 181, 1968.
[7] V. N. Vapnik, *The Nature of Statistical Learning*. New York: Springer-Verlag, 1995.
[8] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.
[9] S. M. Omohundro, "Geometric learning algorithms," *Physica D*, vol. 42, pp. 307–321, 1990.
[10] G. C. Vasconcelos, M. C. Fairhurst, and D. L. Bisset, "Investigating feedforward networks with respect to the rejection of spurious patterns," *Pattern Recognition Lett.*, vol. 16, no. 2, pp. 207–212, 1995.
[11] B. A. Telfer and D. P. Casasent, "Minimum-cost associative processor for piecewise-hyperspherical classification," *Neural Networks*, vol. 6, pp. 1117–1130, 1993.
[12] N. Burgess, "A constructive algorithm that converges for real-valued input patterns," *Int. J. Neural Syst.*, vol. 5, no. 1, pp. 59–66, Mar. 1994.
[13] I. Guyon, B. E. Boser, and V. N. Vapnik, "Automatic capacity tuning of very large VC-dimension classifiers," in *Advances in Neural Information Processing Systems V*, S. J. Hanson, J. D. Cowan, and C. L. Giles, Eds., vol. 5. San Mateo, CA: Morgan Kaufmann, 992.
[14] J. Moody and C. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Computa.*, vol. 1, pp. 281–294, 1989.
[15] S. Ridella, S. Rovetta, and R. Zunino, "Adaptive internal representation in circular backpropagation networks," *Neural Comput. Applicat.*, vol. 3, pp. 222–233, 1995.

[16] F. Rosenblatt, *Principles of Neurodynamics*. New York: Spartan, 1962.
[17] D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing*. Cambridge, MA: MIT Press, 1986.
[18] R. P. Lippmann, "An introduction to computing with neural nets," *IEEE Acoust., Speech, Signal Processing Mag.*, vol. 4, pp. 4–22, 1987.
[19] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 21–27, Jan. 1967.
[20] S. B. Holden and P. J. W. Rayner, "Generalization and PAC learning: Some new results for the class of generalized single-layer networks," *IEEE Trans. Neural Networks*, vol. 6, pp. 368–380, 1995.
[21] Q. Zhang and A. Benveniste, "Wavelet networks," *IEEE Trans. Neural Networks*, vol. 3, pp. 889–898, 1992.
[22] V. N. Vapnik and A. J. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," *Theor. Prob. Applicat.*, vol. 16, pp. 264–280, 1971.
[23] E. B. Baum and D. Haussler, "What size net gives valid generalization?" *Neural Computa.*, vol. 1, pp. 151–160, 1989.
[24] M. Anthony and N. Biggs, *Computational Learning Theory*, Cambridge tracts in theoretical computer science. Cambridge, U.K.: Cambridge Univ. Press, 1992.
[25] R. S. Wenocur and R. M. Dudley, "Some special Vapnik–Chervonenkis classes," *Discrete Math.*, vol. 33, pp. 313–318, 1981.
[26] S.-C. Huang and Y.-F. Huang, "Bounds on the number of hidden neurons in multilayer perceptrons," *IEEE Trans. Neural Networks*, vol. 2, pp. 47–55, Jan. 1991.
[27] E. D. Sontag, "Sigmoids distinguish more efficiently than heavisides," *Neural Computa.*, vol. 1, pp. 470–472, 1989.
[28] ———, "Sigmoids distinguish more efficiently than heavisides," SYCON-Rutgers Center Syst. Contr., Tech. Rep. 89-12, Aug. 1989.
[29] ———, "Shattering all sets of k points in 'general position' requires $(k-1)/2$ parameters," Dep. Math., Rutgers Univ., New Brunswick, NJ, Tech. Rep. 96-01, Feb. 1996.
[30] G. Mirchandani and W. Cao, "On hidden nodes for neural nets," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 661–664, May 1989.
[31] G. J. Mitchison and R. M. Durbin, "Bounds on the learning capacity of some multilayer networks," *Biol. Cybern.*, vol. 60, pp. 345–356, 1989.
[32] A. Kowalczyk, "Counting function theorem for multilayer networks," in *Advances in Neural Information Processing Systems VI*, J. D. Cowan, G. Tesauro, and J. Alspector, Eds.. San Mateo, CA: Morgan Kaufmann, 1993, pp. 375–382.
[33] ———, "Estimates of storage capacity of multilayer perceptron with threshold logic hidden units," *Neural Networks*, to appear.
[34] P. Niyogi and F. Girosi, "On the relationship between generalization error, hypothesis complexity, and sample complexity for radial basis functions," *Neural Computa.*, vol. 8, no. 4, pp. 819–842, May 1996.
[35] J. de Villiers and E. Barnard, "Backpropagation neural nets with one and two hidden layers," *IEEE Trans. Neural Networks*, vol. 4, pp. 136–141, 1993.
[36] R. P. W. Duin, "A note on comparing classifiers," *Pattern Recognition Lett.*, vol. 17, no. 5, pp. 529–536, 1996.
[37] K. Lang and M. Witbrock, "Learning to tell two spirals apart," in *Proc. Connectionist Models Summer School*, 1989.
[38] S. E. Fahlmann and C. Lebiere, "The cascade-correlation learning architecture," in *Advances in Neural Information Processing Systems II*. San Mateo, CA: Morgan Kaufmann, 1989, pp. 524–532.
[39] D. H. Deterding, "Speaker normalization for automatic speech recognition," Ph.D. dissertation, Cambridge Univ., Cambridge, U.K., 1989.
[40] T. P. Vogl, J. K. Mangis, A. K. Rigler, W. T. Zink, and D. L. Alkon, "Accelerating the convergence of the backpropagation method," *Biol. Cybern.*, vol. 59, pp. 257–263, 1988.
[41] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford, U.K.: Clarendon Press, 1995.
[42] S. K. Park and K. W. Miller, "Random number generators: Good ones are hard to find," *Commun. ACM*, vol. 31, no. 10, pp. 1192–1201, Oct. 1988.
[43] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 1992.
[44] M. R. Berthold and J. Diamond, "Boosting the performance of RBF networks with dynamic decay adjustment," in *Advances in Neural Information Processing Systems VII*, G. Tesauro, D. S. Touretzky, and T. K. Leen, Eds. San Mateo, CA: Morgan Kaufmann, 1994, pp. 521–528.

Sandro Ridella (M'93) received the Laurea degree in electronic engineering from the University of Genova, Italy, in 1966.

He is a full Professor in the Department of Biophysical and Electronic Engineering, University of Genova, Italy, where he teaches circuits and algorithms for signal processing. In the last five years, his scientific activity has been mainly focused on the field of neural networks.

Stefano Rovetta received the Laurea degree in electronic engineering from the University of Genova, Italy. In 1993 he joined the Electronic Systems Group of the Department of Biophysical and Electronic Engineering, University of Genova, where he is currently pursuing the Ph.D. degree in models, methods, and tools for electronic and electromagnetic systems.

His research interests include neural networks and electronic circuits and systems.

Rodolfo Zunino (S'90–M'91) received the Laurea degree in electronic engineering from the University of Genova, Italy, in 1985.

From 1986 to 1995 he was a Research Consultant with the Department of Biophysical and Electronic Engineering of Genova University. Since 1995 he is with the same department as an Assistant Professor of Applied Electronics. His interests include distributed systems and neural-network methods.

Mr. Zunino is a Member of AEI.