

# HIGH CAPACITY ASSOCIATIVE MEMORY BASED ON THE RANDOM NEURAL NETWORK MODEL

ARISTIDIS LIKAS and ANDREAS STAFYLOPATIS

*National Technical University of Athens  
Department of Electrical and Computer Engineering  
157 73 Zographou, Athens, Greece*

In this paper the Bipolar Random Network is described, which constitutes an extension of the Random Neural Network model and exhibits autoassociative memory capabilities. This model is characterized by the existence of positive and negative nodes and symmetrical behavior of positive and negative signals circulating in the network. The network's ability of acting as autoassociative memory is examined and several techniques are developed concerning storage and reconstruction of patterns. These approaches are either based on properties of the network or constitute adaptations of existing neural network techniques. The performance of the network under the proposed schemes has been investigated through experiments showing very good storage and reconstruction capabilities. Moreover, the scheme exhibiting the best behavior seems to outperform other well-known associative neural network models, achieving capacities that exceed  $0.5n$  where  $n$  is the size of the network.

*Keywords:* Neural computation, associative memory, random neural network, Hebbian learning, spectral learning.

## 1. INTRODUCTION

Autoassociative neural network models have been extensively studied, especially the Hopfield model<sup>15</sup> which constitutes the most popular network of this type, mainly because of its simplicity and the fact that it can be easily implemented on hardware. However the Hopfield model is characterized by several drawbacks when acting as an autoassociative memory device: low memory capacity (about  $0.15n$ ), many spurious patterns, small radius of attraction around each pattern stored and sensitivity to the properties of the patterns stored.

Several attempts have been made to deal with the above weaknesses. Some of them aimed at improving performance and storage capacity by introducing more sophisticated learning rules, while other attempts have focused on improving the recollection dynamics so as to achieve a better exploitation of the information contained in the weight matrix.<sup>18,21,22</sup> Our work follows the latter direction in the sense that we have developed a different autoassociative model, the Bipolar Random Network, and we examine its performance with respect to other models (such as the Hopfield network or Morita's extensions) using the same incremental learning schemes (Hebbian or spectral<sup>20</sup> learning algorithms).

The proposed associative network model constitutes an extension of the Random Neural Network originally introduced by Gelenbe.<sup>7,8</sup> Although the work on the Random Neural Network was initially motivated by the behavior of natural

neural networks, the model can represent general systems containing processes and resources and supporting some types of control operations. In fact, the model is based on probabilistic assumptions and belongs to the family of Markovian queueing networks. The novelty with respect to usual queueing models lies in the concept of requests for removing work (negative customers) in addition to classical requests for performing work (positive customers). This novel class of models are referred to as *G-networks* in queueing network literature and has been studied extensively during the last few years.<sup>9,11,14</sup>

In the neural network context, the Random Neural Network model has been the basis of theoretical efforts and applications<sup>3,4,19</sup>. An important issue is that the model can be constructed in conformity to usual neural network characteristics. Moreover, the output of the model can be expressed in terms of its steady-state solution, that is in terms of quantities obtained through direct numerical computations.

The Bipolar Random Network studied here is adapted to associative memory operation. An application of this model to the retrieval of symbolic pictures is described in Ref. 19, where the patterns stored were characterized by a special hierarchical structure. In the present paper we provide a systematic description of several schemes that can be developed for the bipolar random network and attempt to evaluate the capacity and error correction capability of each scheme based on experiments with randomly created patterns. Moreover, we introduce an iterative recollection scheme that exhibits the best performance and is characterized by higher capacity indices when compared with other neural associative memories.

In the next section we briefly present the main characteristics of the bipolar random network model. Issues related to the encoding of patterns in the network are addressed in Sec. 3, whereas reconstruction schemes are investigated in Sec. 4. Section 5 deals with the experimental evaluation of the proposed schemes in comparison with other associative memory models. Conclusions are discussed in Sec. 6.

## 2. DEFINITION OF THE MODEL

### 2.1. The Bipolar Random Network

The *Random Neural Network*<sup>7,8</sup> is a model that reproduces the pulsed behavior of natural neural systems. It is based on probabilistic assumptions and is characterized by the existence of signals in the form of spikes of unit amplitude that circulate among nodes. Positive and negative signals represent excitation and inhibition, respectively. The major property of the model is that it accepts a product form solution, i.e., the network's stationary probability distribution can be written as the product of the marginal probabilities of the state of each node. In the applications considered here, we are interested in an extension of the original random network model that exhibits associative memory capabilities.<sup>10,16,19</sup> This extended model includes two types of nodes — *positive* and *negative* — and preserves the main characteristics of the original model, which considers only positive nodes. The significant feature of the model — in both the original and the extended version —

is that it is analytically solvable, and therefore computationally efficient, since its application is reduced to obtaining solutions to a system of fixed-point equations. In the remainder of this subsection we will provide a brief description of the extended version, which will be referred to as the *Bipolar Random Network*. A detailed description of this model, along with analytical derivation of its main properties, can be found in Ref. 19.

In the bipolar random network, each node accumulates signals that either arrive from the outside of the network or from other nodes. External positive and negative signal arrivals to each node  $i$  are considered Poisson with rates  $\Lambda(i)$  and  $\lambda(i)$ , respectively. If the total signal count of a node at a given time instant is strictly positive, the node fires and sends out spikes to other nodes or to the outside of the network. The intervals between successive firing instants at node  $i$  are random variables following an exponential distribution with mean  $1/r(i)$ . Positive (negative) nodes accumulate and emit only positive (negative) signals. The role of negative (positive) signals at positive (negative) nodes is purely suppressive, i.e., they simply cancel signals of the opposite sign if there are any.

Connections between nodes can be positive or negative, so that a signal leaving a node can move to another node as a signal of the same or the opposite sign, respectively. More specifically, a positive (negative) signal leaving positive (negative) node  $i$  arrives to node  $j$  as a positive (negative) signal with probability  $p^+(i, j)$  and as a negative (positive) signal with probability  $p^-(i, j)$ . Also, a signal leaving node  $i$  departs from the network with probability  $d(i)$ . Obviously, for a network with  $n$  nodes we shall have  $\sum_{j=1}^n [p^+(i, j) + p^-(i, j)] + d(i) = 1$ , for  $i = 1, \dots, n$ .

As already stated, the above described Markovian network has product form solution. This property has been shown in Ref. 7 for the original version of the random neural network and in Ref. 19 for the extended version including positive and negative nodes. Results concerning the existence and uniqueness of the solution can be found in Ref. 8. Additional properties of the random neural network model have been established in Refs. 12 and 13.

The steady-state characteristics of the bipolar random network can be summarized as follows.<sup>19</sup> Considering a network with  $n$  nodes its stationary probability distribution is given by  $p(\hat{k}) = \lim_{t \rightarrow \infty} \text{Prob}[\hat{K}(t) = \hat{k}]$ , whenever this limit exists, where  $\hat{K}(t)$  is the state vector at time  $t$  representing the signal count at each node of the network, and  $\hat{k} = (k_1, \dots, k_n)$  denotes a particular value of the vector. Since both types of nodes accumulate signals of the corresponding sign only, an unsigned value  $k_i$  is sufficient for describing the state of a node  $i$ , once the sign of the node is given. If we denote by  $\mathcal{P}$  and  $\mathcal{N}$  the sets of positive and negative nodes, respectively, the flow of signals in the network can be described by the following equations in terms of the arrival rates  $\gamma^+(i)$  and  $\gamma^-(i)$  of positive and negative signals to node  $i$ :

$$\gamma^+(i) = \Lambda(i) + \gamma_{int}^+(i) = \Lambda(i) + \sum_{j \in \mathcal{P}} q_j r(j) p^+(j, i) + \sum_{j \in \mathcal{N}} q_j r(j) p^-(j, i) \quad (1)$$

$$\gamma^-(i) = \lambda(i) + \gamma_{int}^-(i) = \lambda(i) + \sum_{j \in \mathcal{P}} q_j r(j) p^-(j, i) + \sum_{j \in \mathcal{N}} q_j r(j) p^+(j, i) \quad (2)$$

where  $\gamma_{int}^+(i)$ ,  $\gamma_{int}^-(i)$  represent the internal arrival rates of positive and negative signals respectively, and

$$q_i = \frac{\gamma^+(i)}{r(i) + \gamma^-(i)}, \quad i \in \mathcal{P} \quad (3)$$

$$q_i = \frac{\gamma^-(i)}{r(i) + \gamma^+(i)}, \quad i \in \mathcal{N} \quad (4)$$

It can be shown<sup>19</sup> that, if a unique non-negative solution  $\{\gamma^+(i), \gamma^-(i)\}$  exists in relation to the above equations such that  $q_i < 1$ , then the steady-state network probability distribution has the form:

$$p(\hat{k}) = \prod_{i=1}^n [1 - q_i] q_i^{k_i} \quad (5)$$

Clearly,  $q_i$  is the steady-state probability that node  $i$  is excited and is the key quantity associated with application of the model. It represents the level of excitation as an analog rather than as a binary variable, thus leading to a more detailed information on system state. In a sense, each node acts as a non-linear frequency demodulator, since it transforms the frequency of incoming spike trains into an “amplitude” value  $q_i$ . We take advantage of this feature when employing the steady-state solution of the network. An analytical solution of the set of Eqs. (1)–(4) is not always feasible. However, there exist cases of networks, for which the existence of a solution can be proved.<sup>8</sup> In the work presented here, standard numerical analysis techniques are used for the solution of the non-linear system of equations.

## 2.2. Associative Memory Operation

Our aim is to use the bipolar random network as the basis for autoassociative memory operation in a manner analogous to relevant neural network models, such as the Hopfield network. In general, the construction of a bipolar random network requires specification of the following characteristics:

- Routing probabilities and firing rates
- External arrival rates
- Sign of nodes

An analogy between usual neural network representation and the random network model can be constructed.<sup>7</sup> Each neuron is represented by a node of the random network. Considering the non-output neuron  $i$ , the parameters of the corresponding node  $i$  are chosen as follows:

$$d(i) = 0 \quad (6)$$

$$r(i)p^+(i, j) = w_{ij}, \quad \text{if } w_{ij} > 0 \quad (7)$$

$$r(i)p^-(i, j) = |w_{ij}|, \quad \text{if } w_{ij} < 0 \quad (8)$$

where  $w_{ij}$  is the connection weight from neuron  $i$  to neuron  $j$ . Summing over all  $j$ , the firing rate  $r(i)$  of the non-output node  $i$  is set

$$r(i) = \sum_j |w_{ij}| \quad (9)$$

Finally, for each output node  $i$ , we set  $d(i) = 1$  and some appropriate value is assigned to  $r(i)$ . Although the above quantities represent rates (or frequencies) of spike emission, they clearly play a role similar to that of synaptic weights.

In what follows, we will restrict our interest to networks constructed according to the following assumptions:

1. All nodes are non-output ( $d(i) = 0$  for each  $i$ ).
2. External signals arriving to each node  $i$  have the same sign as this node, i.e. either  $\Lambda(i) \neq 0$  and  $\lambda(i) = 0$  or  $\Lambda(i) = 0$  and  $\lambda(i) \neq 0$ .

Since all nodes are non-output, the choice of routing probabilities and firing rates can also be expressed in terms of a matrix  $W$  of connection weights by using Eqs. (7)–(9). Once the connection weights  $w_{ij}$  have been determined, the computation of routing probabilities and firing rates is straightforward. Also we denote by  $\hat{x} = (x_1, \dots, x_n)$  the vector representing the sign of nodes, i.e.  $x_i = 1$  if node  $i$  is positive, otherwise  $x_i = -1$ . Therefore, the specification of the sign vector can be viewed as the application of a bipolar pattern to the network. Upon this notion of applying a pattern as input to the network is based on the operation of the network as an associative memory. More specifically, an *input pattern* (also called a probe pattern) constitutes a partial or distorted version of one of a number of fundamental patterns that are stored or memorized by the network. The result of the network's operation is the correction of the input pattern or equivalently reconstruction of the respective stored pattern (also referred to as retrieval or recollection). The corrected version of the input pattern will be considered as the *output* of the network.

Let  $\{\hat{x}^1, \dots, \hat{x}^m\}$  be  $m$  bipolar patterns to be stored in a bipolar random network with  $n$  nodes and  $x_i^k \in \{-1, 1\}$  denoting the  $i$ th component of pattern  $\hat{x}^k$ . In order to store these patterns, first the connection weights  $w_{ij}$  must be determined according to some learning rule and then the parameters  $p^+(i, j)$ ,  $p^-(i, j)$ ,  $r(i)$  will be computed using Eqs. (7)–(9). Therefore, using a learning rule, the *fixed part* (routing probabilities and firing rates) of a bipolar random network is specified.

There is also the *variable part* which consists of the sign of nodes and the external signal arrival rates. This variable part is determined during operation by the input pattern that is applied each time to the random network. In particular, if the  $i$ th element of the input pattern is equal to 1, then the  $i$ th node of the network is set positive and the arrival rates of positive and negative signals to this node are  $\Lambda(i) \neq 0$  and  $\lambda(i) = 0$ , respectively. If the  $i$ th element of the input pattern is equal to  $-1$ , then the  $i$ th node is set negative with  $\Lambda(i) = 0$  and  $\lambda(i) \neq 0$ .

Each time an input pattern is applied to the network during operation, the numerical solution of the nonlinear system (1)–(4) yields the quantities  $q_i$ ,  $\gamma_{int}^+(i)$ ,

$\gamma_{int}^-(i)$  for  $i = 1, \dots, n$ . Then a correction procedure takes place which consists of identifying the wrong components of the input pattern based on information provided by those values.

### 3. ENCODING OF BIPOLAR PATTERNS

A general remark concerning the learning algorithms that are used to train autoassociative networks is that their objective is to establish a *characteristic* property of the patterns stored. Such a property must be "supported" by the operation of the network, in the sense that the network should be able to provide output patterns satisfying the property. Moreover, it is desirable that after the learning phase only the patterns already stored possess the above characteristic so that they can be identified during recollection and distinguished from other patterns.

In what concerns the bipolar random network we are interested in *incremental* learning algorithms. Such algorithms have the property that, each time a new pattern must be stored in the network, the update of weights is performed based on information related to the current weight values and to the pattern to be stored, without needing to reconsider the patterns already stored.

#### 3.1. Learning Algorithms based on Stability

Two simple incremental learning algorithms that have been used in our experiments are the *outer-product* rule and the *spectral* rule. Both learning schemes have been developed in the context of the Hopfield neural network and are characterized by their ability to provide (to some extent) the patterns stored with the *stability property*. This means that if the network has attained a state corresponding to a pattern stored, no transition out of this state can occur according to the dynamics of the network.<sup>15</sup> More specifically, in order for a stored pattern  $\{\hat{x}^1, \dots, \hat{x}^m\}$  to be a stable state of a Hopfield network with weights  $w_{ij}$ , the following condition must be true<sup>1,15,17</sup>:

$$\text{sign}\left(\sum_{j=1}^n x_j^k w_{ji}\right) = \text{sign}(x_i^k), \quad i = 1, \dots, n, \quad k = 1, \dots, m \quad (10)$$

The above requirement can also be written as

$$x_i \left[ \sum_j x_j w_{ji} I(x_j w_{ji} > 0) - \sum_j |x_j w_{ji}| I(x_j w_{ji} < 0) \right] > 0, \quad (11)$$

$$i = 1, \dots, n, \quad k = 1, \dots, m$$

where the indicator function  $I(X)$  is 1 if  $X$  is true and 0 otherwise.

In the outer-product case the weights are computed according to the simple Hebbian rule:

$$w_{ij} = \sum_{k=1}^m x_i^k x_j^k, \quad i = 1, \dots, n, \quad j = 1, \dots, n, \quad i \neq j \quad (12)$$

$$w_{ii} = 0, \quad i = 1, \dots, n \quad (13)$$

If the Hebbian rule is used for constructing the connection matrix  $W$ , the stability property of each pattern stored holds with high probability, especially when the size of the network is significantly larger than the number of patterns.<sup>2,5,17</sup> In general, we cannot guarantee that the stability property holds for every pattern  $\hat{x}^k$  (especially when  $m$  has a magnitude comparable to  $n$ ), except for the case where all patterns are orthogonal. Orthogonality can be established with high probability when the patterns  $\hat{x}^k$  are generated from a sequence of symmetric Bernoulli trials. In this case, the capacity of the outer-product scheme is  $m = O(n/\log n)$ .<sup>17</sup>

The *spectral scheme*<sup>20</sup> guarantees that each pattern  $\hat{x}^k$  will be a stable state of a Hopfield network provided that the  $m$  patterns to be stored are linearly independent. It seems to exhibit better performance than the outer-product scheme (capacity  $m = O(n)$ ), but the computational cost of constructing the interconnection matrix  $W$  is higher. Several "spectral" formulations of the matrix  $W$  have been proposed.<sup>6,20</sup> We have considered the following incremental computation scheme. Once the patterns  $\hat{x}^l$ ,  $1 \leq l \leq k-1$ , have already been stored, to store the pattern  $\hat{x}^k$  the current value of the weight matrix becomes:

$$W[k] = W[k-1] + \frac{e^k e^{k\top}}{\hat{x}^{k\top} e^k}, \quad k = 1, \dots, m \quad (14)$$

with  $W[0] = 0$  and  $e^k = (\lambda I - W[k-1])\hat{x}^k$ . The parameter  $\lambda$  was set equal to  $n$  in our experiments.

The above learning schemes that establish stable patterns in the context of the Hopfield network, can also be used for storing patterns in a bipolar random network. It has been experimentally observed that when one of the patterns stored is presented as input to a random network that has been trained using the above rules, the resulting  $q_i$  values are relatively high and close to each other. On the contrary when an erroneous pattern is presented as input to the network, the  $q_i$  values corresponding to erroneous nodes are in general lower than the  $q_i$  values corresponding to correct nodes. This characteristic can be used for identification of wrong components. Another characteristic of the patterns stored that is supported by the above learning algorithms and is closely related to the operation of the bipolar random network is the property of *consistency* defined next.

### 3.2. The Consistency Property

**Definition 1.** We shall say that a node of the bipolar random network is *consistent* (with its sign) if, in the steady-state, the internal arrival rate of signals it accepts exceeds the internal arrival rate of signals it rejects:

$$x_i(\gamma_{int}^+(i) - \gamma_{int}^-(i)) > 0 \quad (15)$$

**Definition 2.** We define the *degree of consistency* of each node of a bipolar random network as the quantity:

$$\beta_i = x_i(\gamma_{int}^+(i) - \gamma_{int}^-(i)) \quad (16)$$

Clearly, nodes with positive degree of consistency are consistent, as defined above.

**Definition 3.** A bipolar random network is called *consistent* if all its nodes are consistent with their sign. A bipolar pattern  $\hat{x} = (x_1, \dots, x_n)$ , that is presented as input to a given bipolar random network, is called a *consistent pattern* with respect to the network if it makes the network consistent.

The interpretation of the above definitions is that when a consistent pattern is presented as input, the resulting network has the property that the sign of each node is consistent with its input rate from the other nodes of the network.

The consistency property of a given pattern can also be expressed as:

$$x_i \left[ \sum_j q_j x_j w_{ji} I(x_j w_{ji} > 0) - \sum_j q_j |x_j w_{ji}| I(x_j w_{ji} < 0) \right] > 0, \quad i = 1, \dots, n \quad (17)$$

It is important to notice the similarity between the definition of a stable pattern in the context of the Hopfield neural network and the definition of a consistent pattern in the context of the bipolar random network.<sup>16</sup> In fact, as Eqs. (11) and (17) indicate, if all the  $q_i$  values are equal to each other for a given network, then the two properties become equivalent, in the sense that if one of them holds for a given node, the other will hold too. In the Hopfield model, the asynchronous update algorithm is based on checking whether a selected node satisfies the stability property. The same principle can be applied in order to develop analogous algorithms for the bipolar random network that exploit the consistency property.

#### 4. CORRECTION PROCEDURES

As mentioned in the previous section, a first general approach for defining the correction procedure can be based on the fact that the solution of the flow equations yields low and high  $q_i$  values for nodes corresponding to wrong and correct components of the input pattern, respectively. This means that the wrong components of the input pattern are expected to correspond to the lowest  $q_i$  values. In fact, experimental results show that the network clearly tends to enhance the distance between  $q_i$  values which are supposed to indicate wrong and correct components, but this differentiation is not always strong enough and can vary following the input pattern. Hence, it is not easy to define an appropriate threshold value that could help us perform the discrimination.

Instead, we can successively correct the sign of components of the output pattern  $\hat{y}$  starting from the lowest  $q_i$  value and following an increasing  $q_i$  value order, until a good reconstruction is obtained. Since the number of wrong components is generally not known *a priori*, we need a means of estimating the quality of the reconstruction in order to stop the correction procedure. Estimating the quality of the reconstruction is equivalent to being in a position to identify a stored pattern. Therefore, the characteristic properties discussed previously are very helpful in this direction. If an output pattern is found that satisfies such a property, then we can consider that a stored pattern has been reconstructed, thus, the recall should terminate. Otherwise, the procedure should continue by correcting the sign of the component corresponding to the immediately succeeding  $q_i$  value.



In general, it is possible to apply one of the following two strategies: a *direct* strategy, where the non-linear system is solved only once for the initially provided input pattern and the choice of the nodes that should be updated is based on the resulting  $q_i$  values; an *iterative* strategy, in which the non-linear system is solved each time the sign of one or more nodes is corrected, in order to find the  $q_i$  values that correspond to the new sign vector, and reconstruction continues based on the updated  $q_i$  values. It is obvious that the iterative strategy is more effective than the direct one, but it is more expensive in terms of the required computational effort. The schemes presented next follow either of the above general directions.

#### 4.1. Direct Correction

Direct correction schemes can be developed that are based either on the stability or the consistency property.

When the Hebbian rule or the Spectral learning scheme is used for computing the matrix  $W$  of a bipolar random network, the stability property is satisfied by the patterns stored with high probability as stated previously. We should note here, that the stability property must be viewed as a characteristic of stored patterns expressed mathematically by (10). This property is due to the learning scheme and is not related to the operation of the network, as is the case with the Hopfield network. Being a characteristic of stored patterns the stability property can be used for their identification. Thus, during the correction procedure described previously, each candidate output pattern  $\hat{y} = (y_1, \dots, y_n)$  is associated with a counter, whose value is equal to the number of indices  $i$  that do not satisfy the equation

$$\text{sign}\left(\sum_{j=1}^n y_j w_{ji}\right) = \text{sign}(y_i) \quad (18)$$

The pattern with zero counter or minimum counter after a predefined maximum number of corrections is taken as the output pattern.

In the above scheme, the analog quantities  $q_i$  are simply used to imply erroneous nodes. Indeed, the stability property, used as a criterion for identification, involves only sign information. On the other hand, the consistency property involves both the signs of nodes and the information provided by the quantities  $q_i$ . Therefore, a direct scheme based on the consistency property cannot be applied in the same manner, since checking whether each candidate output pattern  $\hat{y}$  is consistent, requires the non-linear system to be solved again. This is what happens in the iterative asynchronous schemes described in the following subsections.

Nevertheless, a reasonable heuristic would be to apply the consistency criterion to each node separately; thus there is an indication for wrong components and correction can be obtained in a straightforward manner. More specifically, when a noisy version of a pattern stored is presented as input to the network, we solve the non-linear system of equations and compute the analog vector  $\hat{q} = (q_1, \dots, q_n)$ . Then we recognize the erroneous nodes based on the notion of consistency, by calculating the quantities  $\gamma_{int}^+(i)$ ,  $\gamma_{int}^-(i)$  and checking whether condition (15) holds

for each  $i = 1, \dots, n$ . In case a node  $i$  does not satisfy the above condition, we can consider that there exists an error in the component  $i$  of the input pattern. Thus, we can identify erroneous nodes and reconstruct the noisy input pattern. This method is very simple and fast and provides identification and correction based on information that is local to each node of the network.

The above described direct correction schemes have been introduced in Ref. 19 where the bipolar random network was employed in an application of symbolic picture storage and retrieval. They are included here for reasons of completeness and comparison with the iterative schemes introduced next.

#### 4.2. Iterative Correction

In the synchronous case described previously, once the analog vector  $\hat{q}$  has been computed, all nodes are simultaneously examined for satisfaction of the consistency criterion. In this way, the effect caused on other nodes by the change of the sign of a node is neglected. To take this effect into account, an iterative correction scheme is needed, where the non-linear system is solved each time the sign of a node changes during correction.

More specifically, we start by providing a noisy pattern as input to the network and find the corresponding analog vector  $\hat{q}$  through the solution of the non-linear system of equations. Then at each step of the method, a node is selected and checked for consistency using Eq. (15). If this is the case, no changes will occur, otherwise the sign of node  $i$  will change and the new vector  $\hat{q}$  corresponding to the updated network will be computed. By proceeding in this fashion, a state is reached (in most cases) that corresponds to a consistent network and no further sign change is possible. Since oscillations may appear in some cases, an upper bound (taken equal to  $4n$  in our experiments) can be imposed on the number of iterations that the network is allowed to perform.

In what concerns the node to be selected each time for update there are several options. One of them is to select the node with the lowest  $q_i$  value since this node has the greater probability of being inconsistent. An alternative is to randomly select a node. The latter scheme has been adopted in our experiments since it makes the operation of the bipolar random network totally analogous to that of the Hopfield network, with stability checking being replaced by consistency checking.

An interpretation of the operation of the bipolar random network can be obtained by considering the effect of the  $q_i$  values computed at each iteration. For a given sign vector, the value  $q_i$  provides an indication of the degree to which node  $i$  is active or not. In contrast to the Hopfield case, the decision of whether a node should change its sign or not depends not only on the sign of the other nodes to which it is connected, but also on the degree to which these nodes are active (either positively or negatively). A node corresponding to a noisy component of the input vector, will generally have a low  $q_i$  value (and a low degree of consistency). Thus, the effect of this node to the input rate of the other nodes is reduced and the correction decisions are based mainly on the nodes which are active enough. Indeed,

the role of highly active nodes is most significant for the recollection behavior of the model, as will be discussed next.

#### 4.3. Two-scale Iterative Correction

As suggested in Ref. 18, the capacity of the autocorrelation (Hopfield) model can be increased by modifying the conventional network dynamics so that the influence of neurons which are strongly excited (if they are positive) or strongly inhibited (if they are negative) is reduced. This implies an operation scheme that involves two phases at each step. During the first phase the conventional Hopfield dynamics is used and the neurons that are strongly excited (or inhibited) are identified. During the second phase the percentage of total excitation (or inhibition) of each neuron that is due to the strong neurons is computed and the outputs of some neurons are reversed in case this percentage is significant.

The above procedure is referred to as the *partial reverse method*<sup>18</sup> and is applied to discrete Hopfield networks with discrete-time dynamics in an attempt to enhance the recollection capability of the network by moving the state trajectory away from spurious patterns. Indeed, it is shown in Ref. 18 that the influence of largely active neurons increases the overlapping of the network state with stored patterns other than the pattern of interest. These patterns attract the network state, eventually making it deviate and retreat from the target pattern. This leads to the idea of helping the network avoid local minima by reducing the influence of strong neurons, thus making the network state distant from patterns other than the target. The analog version of the partial reverse method, which is based on neurons with nonmonotone output function, has better recollection capabilities<sup>18,22</sup> and is considered a successful scheme for continuous-time network models.

Following the above principle, we have developed an iterative two-phase operation scheme for the bipolar random network. At each iteration step the non-linear system is solved twice, once for the whole network and a second time for a subnetwork of the original network. This two-scale approach is based on the identification of nodes that are *strongly consistent* and the exclusion of these nodes from the computations of the second phase. The proposed scheme takes into account not only the sign but also the degree of consistency that characterizes each node. The exploitation of this additional information leads to a correction scheme that exhibits significantly improved reconstruction performance. In what concerns the number of nodes that are corrected at each iteration, it can be considered as an intermediate approach between the direct scheme where all inconsistent nodes change sign synchronously and the previous one-scale iterative scheme where a single node is examined at each iteration.

**Definition 4.** We shall say that a node  $i$  of the bipolar random network is *strongly consistent* if its consistency degree  $\beta_i$  exceeds a positive threshold value  $\rho_i^+ > 0$ :

$$x_i(\gamma_{int}^+(i) - \gamma_{int}^-(i)) > \rho_i^+ \quad (19)$$

**Definition 5.** We shall say that a node  $i$  of the bipolar random network is *strongly*

*inconsistent* if its consistency degree  $\beta_i$  is lower than a negative threshold value  $\rho_i^- < 0$ :

$$x_i(\gamma_{int}^+(i) - \gamma_{int}^-(i)) < \rho_i^- \quad (20)$$

In our implementation, the values of  $\rho_i^+$ ,  $\rho_i^-$  are taken equal for all  $i$  and are computed at each iteration as follows:

$$\rho^+ = \frac{\bar{\beta}^+ + \beta_{\max}^+}{2} \quad (21)$$

$$\rho^- = \frac{\bar{\beta}^- + \beta_{\min}^-}{2} \quad (22)$$

where  $\bar{\beta}^+ > 0$  is the average degree of consistency computed over all  $i$  with  $\beta_i > 0$  and  $\beta_{\max}^+ = \max(0, \max_i \beta_i)$ . Similarly,  $\bar{\beta}^- < 0$  is the average degree of inconsistency computed over all  $i$  with  $\beta_i < 0$  and  $\beta_{\min}^- = \min(0, \min_i \beta_i)$ .

The following operations take place at each iteration step of the proposed correction procedure.

- First the non-linear system is solved and, based on the resulting  $q_i$  values, the *strongly consistent* nodes are identified.
- The subnetwork that results from the removal of the strongly consistent nodes is considered and the corresponding non-linear system is solved to find the new  $q_i$  values of the subnetwork nodes. Based on the resulting  $q_i$  values the *strongly inconsistent* nodes are found and their sign is changed. If no such nodes are detected, convergence has been attained, otherwise we proceed to a new iteration step.

As is the case with the partial reverse method, the above procedure does not always converge and frequently exhibits oscillation problems. In order to deal with this problem we have employed a heuristic that is based on the following remark. We have observed that in most reconstruction experiments, all erroneous nodes were contained in the set of inconsistent nodes (not necessarily strongly inconsistent) identified during the second phase of the *first* iteration. To exploit this observation, at the first iteration step we *mark* the nodes that belong to the above set and our task then is to distinguish which of these nodes correspond to erroneous components in order to change their sign. Therefore, during the second phase of each iteration, only the strongly inconsistent nodes that are *marked* have permission to change their sign.

The employment of the above heuristic has the advantage that the search for erroneous nodes is confined to the subset containing the marked network nodes. Moreover, the oscillation problem is reduced, since the number of nodes that are allowed to change their sign at each iteration is smaller. Convergence is attained, if at some iteration no marked nodes exist that are strongly inconsistent. Since it is not possible to prove guaranteed convergence, an upper bound must be imposed on the number of iterations that the network is allowed to perform (which was taken

equal to 10 in our experiments). The sign vector at the iteration step, at which this upper bound is exceeded, is considered as the recollection of the network.

## 5. EXPERIMENTAL RESULTS AND PERFORMANCE ISSUES

The four operation schemes described in the previous section (two direct and two iterative) were tested. For reasons of comparison we have also tested the performance of the Hopfield model and the Morita partial reverse model using exactly the same patterns. For each scheme both the Hebbian and the spectral learning rules were employed. The partial reverse method was not tested under spectral learning, since it is closely related to autocorrelation (Hebbian) storage and using another learning scheme would require adjustment of model parameters as given in Ref. 18 (a non-trivial intervention to the model).

We were interested in assessing the capacity and error correction capability of the various operation schemes for several values of the network size  $n$  (in the range between 20 and 200) and under several levels of distortion  $d$ , as explained below. In particular we have considered four values of  $d$ , namely 0.1, 0.15, 0.2 and 0.25.

Experiments for given values of  $n$ ,  $m$  and  $d$  were conducted as follows. First the  $m$  patterns to be stored in the network were generated by a sequence of symmetric Bernoulli trials (each component was set to 1 with a probability 0.5, otherwise it was set to -1). Then, the weights of the network were computed accordingly using either the Hebbian or the spectral formulation. During operation, 10 experiments were performed for each stored pattern. In each experiment, the stored pattern was randomly distorted to the given distortion level  $d$  (by flipping the state of  $dn$  nodes selected at random) and the resulting erroneous pattern was presented as input to the random network (applying each of the four operation schemes described previously), as well as the Hopfield network and the partial reverse model. The recall for a given operation scheme was considered successful, if *perfect reconstruction* of the original pattern stored was achieved. Thus, for given values of  $n$ ,  $m$  and  $d$ ,  $10m$  reconstruction experiments were performed, and the percentage of correct recollections was computed. In the case where this percentage was greater than 85%, we considered that the network "accepted"  $m$  patterns under distortion level  $d$ . For specific values of  $n$  and  $d$ , the maximum value  $M(n, d)$  of  $m$  for which the network exhibits acceptable performance is used to determine the *storage capacity* of the network under the selected operation scheme. We have adopted this definition, which is close to the most commonly considered notion of capacity, but is also specific in what concerns distortion of input patterns and reconstruction requirements and, thus, well-suited for systematic experimentation and comparative evaluation.

Figures 1-4 illustrate the storage capacity of the various schemes as a function of the size of the network under Hebbian and spectral learning for two different levels of distortion. It can be readily verified that a much higher performance is generally achieved under the spectral learning algorithm. This happens because on the one hand the spectral scheme seems to increase the reconstruction capability of the network, and on the other hand it also increases the reliability during identification.

In all cases, the direct correction schemes feature the lowest capacity (as was more or less expected), whereas the one-scale iterative correction scheme and the Hopfield model have comparable capacity especially under spectral learning. The partial reverse method is superior to the above, but is clearly inferior to the two-scale iterative scheme under Hebbian encoding (Figs. 1 and 2). The latter exhibits an even better performance under spectral encoding (Figs. 3 and 4) achieving a capacity larger than  $0.5n$  for a 10% distortion.

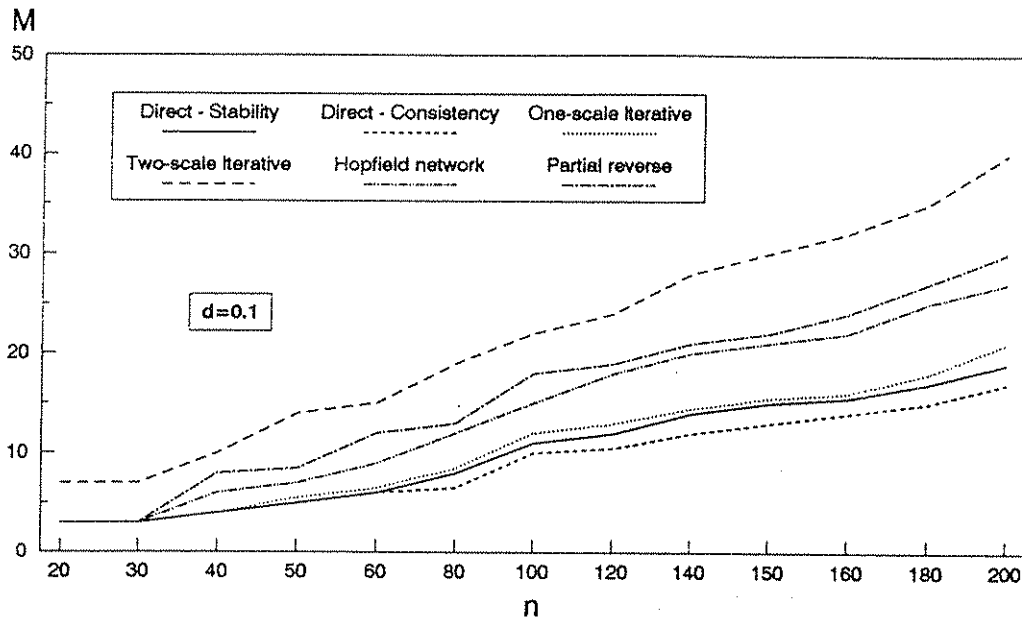


Fig. 1. Storage capacity for 10% distortion (Hebbian learning).

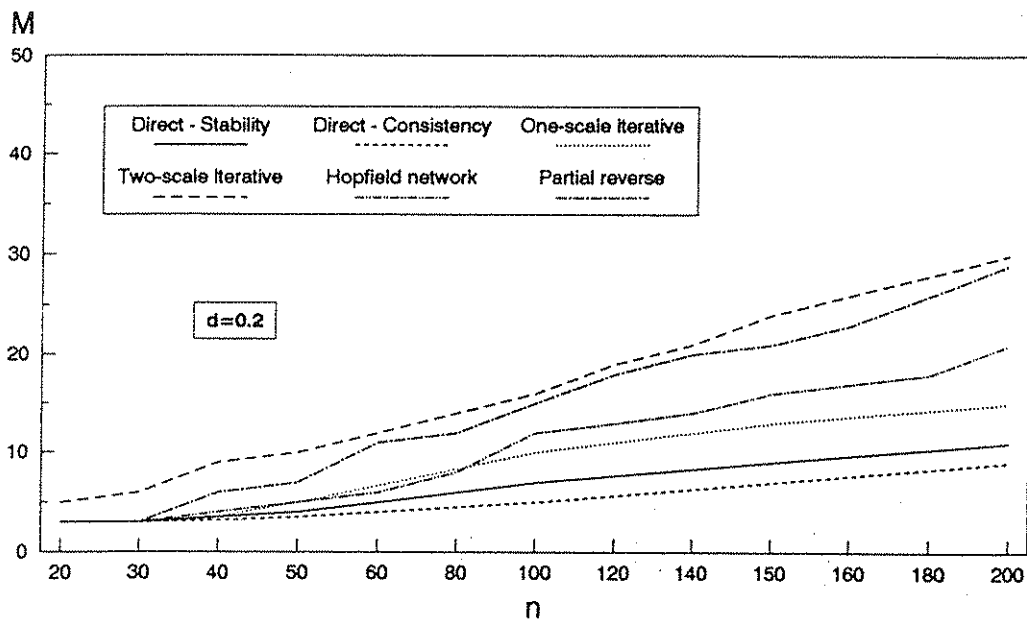


Fig. 2. Storage capacity for 20% distortion (Hebbian learning).

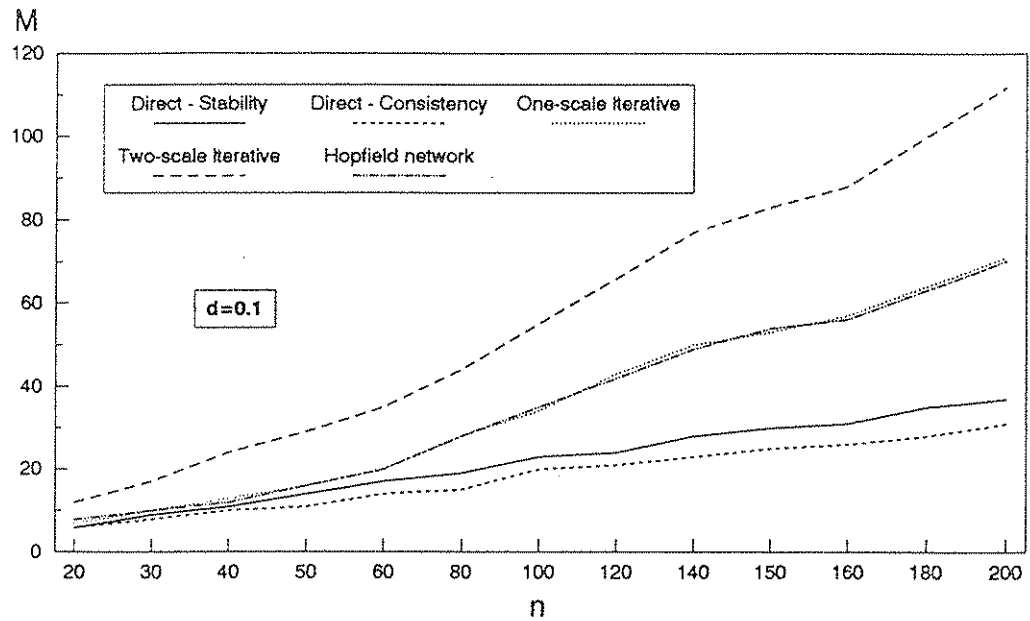


Fig. 3. Storage capacity for 10% distortion (Spectral learning).

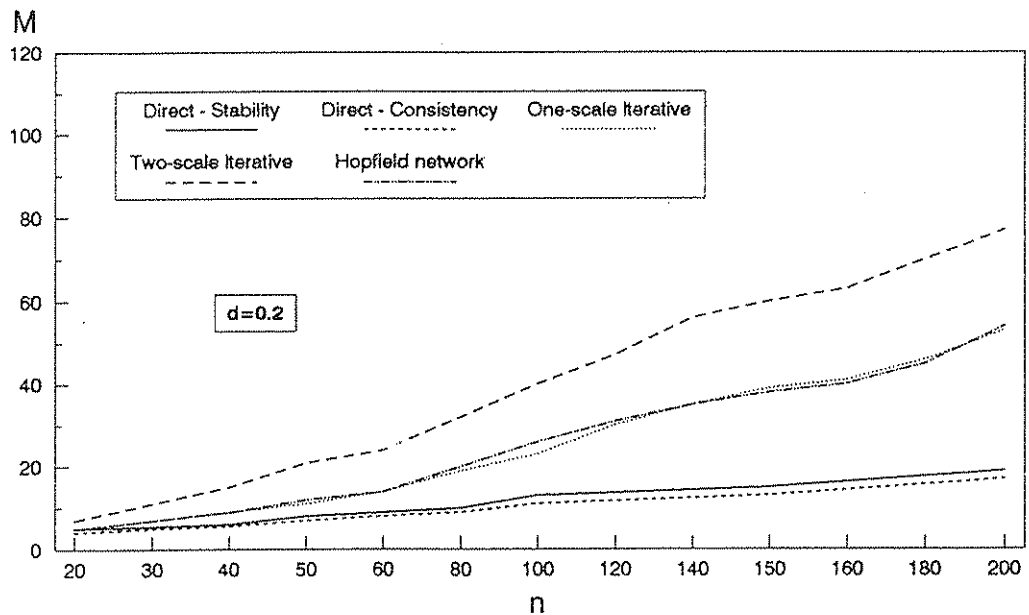


Fig. 4. Storage capacity for 20% distortion (Spectral learning).

Figures 5 and 6 display the success rate, i.e. the percentage of correctly retrieved patterns, as a function of the number of patterns stored in a network of large size ( $n = 200$ ). The three schemes yielding the best capacity are considered, namely the Hopfield network, the partial reverse method and the two-scale iterative scheme of the bipolar random network. Both Hebbian and spectral learning is considered (only Hebbian for the partial reverse model). As can be observed, the two-scale iterative method is by far superior and achieves a 100% success rate under spectral learning (Fig. 6) for as many as  $0.4n$  patterns with a 15% distortion.

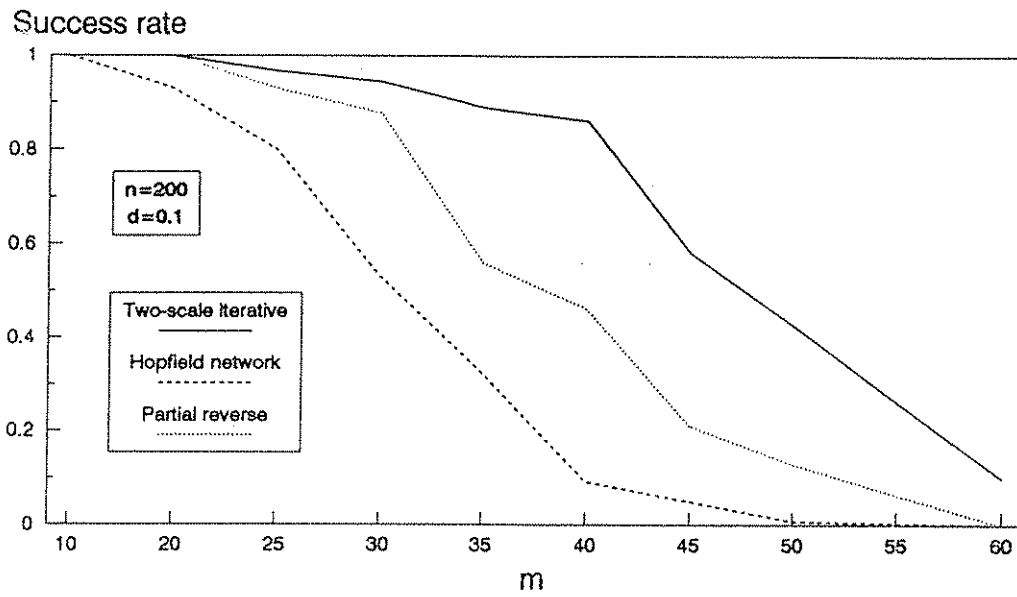


Fig. 5. Success rate (Hebbian learning).

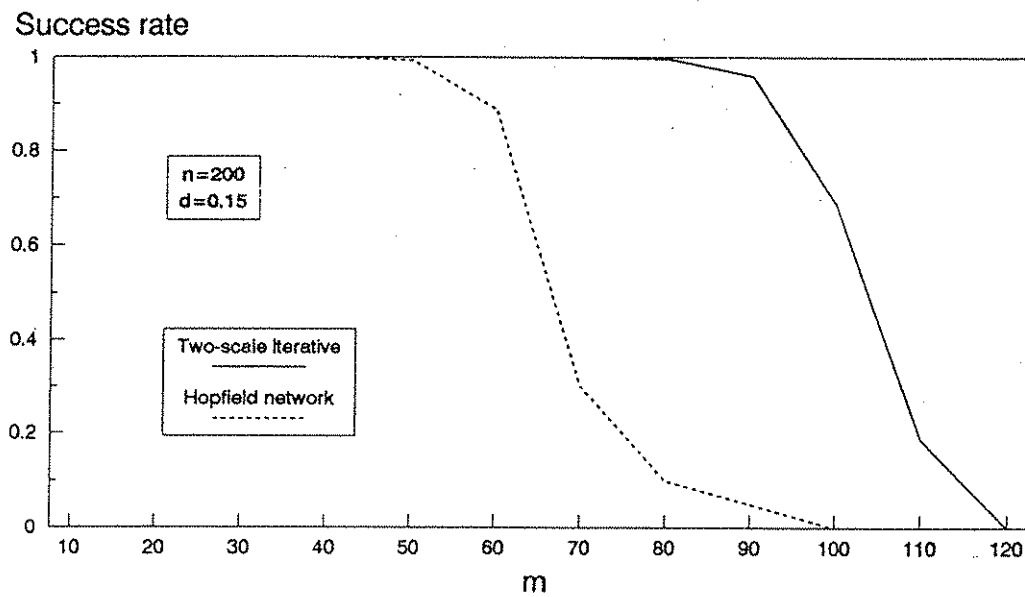


Fig. 6. Success rate (Spectral learning).

Table 1. Capacity as a function of distortion for  $n = 100$  (Hebbian learning).

$d$	Storage capacity		
	Two-scale iterative	Hopfield network	Partial reverse
0.10	22	15	18
0.15	20	14	17
0.20	16	12	15
0.25	14	10	13



Table 2. Capacity as a function of distortion for  $n = 100$  (Spectral learning).

$d$	Storage capacity	
	Two-scale iterative	Hopfield network
0.10	55	35
0.15	48	30
0.20	40	26
0.25	31	19

Finally, Tables 1 and 2 display the variation of storage capacity as a function of the distortion level for the best performing schemes, as before. The two-scale iterative scheme clearly outperforms the other two at all levels of distortion. It should be noted, however, that the relative improvement with respect to the partial reverse model (under Hebbian learning) decreases as distortion grows, which means that the latter method is more robust to noise. Nevertheless, as shown in Table 2, the capacity figures obtained by the two-scale scheme under spectral learning are very satisfactory even for high levels of distortion.

A few remarks are in place concerning the performance of the proposed two-scale iterative scheme. The high storage capacity achieved through this scheme can be attributed to a successful adaptation of ideas improving the recollection dynamics of autocorrelation models to the specific characteristics of the random network model. Our experiments have shown that the use of nonconventional dynamics (such as techniques analogous to the partial reverse method), that improve attractor-like behavior and help avoid spurious memories, can be very effective when combined with the information contained in the random network. Indeed, the notion of consistency involving both analog and discrete information can be the basis for developing powerful recollection algorithms. Relative theoretical work concerning mathematical analysis of the proposed approaches would be very helpful in this direction.

The random network model is an approach based on probabilistic assumptions that features significant biological relevance, since it aims at reproducing the pulsed behavior of natural neural systems. The associative memory techniques considered here are built upon results concerning the steady-state behavior of the network. Although this leads to numerical procedures based on the solution of a non-linear system of equations, the potential of the resulting techniques is closely related to notions with clear physical interpretation (such as activation and consistency) that reflect the properties of the underlying natural paradigm. A possible drawback of the proposed schemes is that the solution of non-linear systems makes them computationally intensive for large-sized problems. It should be noted, however, that the numerical computations involved are easily parallelizable and, hence, the method could greatly benefit from implementation on parallel hardware.

## 6. CONCLUSION

In this paper we have studied the performance of the Bipolar Random Network, which constitutes an adaptation of the Random Neural Network model developed by Gelenbe. In the bipolar network model two types of signals are considered and two types of nodes with opposite behavior. This symmetry seems to be well adapted to the mechanisms involved in associative memory operation. In fact, this network model naturally fits the autoassociative memory paradigm and, in some sense, constitutes a generalization of the Hopfield neural network model in that it exploits both analog ( $q_i$  values) and discrete (sign of nodes) information during recall.

Several schemes related to the learning and recall phases can be devised. The learning performance and error correction capability of the model under such schemes have been investigated in a systematic way through various experiments. Experiments have also been conducted allowing a comparison of the bipolar random network with other neural associative memory models such as the discrete Hopfield network and Morita's partial reverse model. The results obtained were very promising and, in particular, a reconstruction scheme has been proposed that seems to provide better storage capacity than other well-known associative neural network models.

## REFERENCES

1. Y. S. Abu-Mostafa and J.-M. St. Jaques, "Information capacity of the Hopfield model", *IEEE Trans. on Information Theory* **31** (1985) 461-464.
2. S. V. B. Aiyer, M. Niranjana and F. Fallside, "A theoretical investigation into the performance of the Hopfield model", *IEEE Trans. on Neural Networks* **1** (1990) 204-215.
3. V. Atalay, E. Gelenbe and N. Yalabik, "The random neural network model for texture generation", *Int. J. of Pattern Recognition and Artificial Intelligence* **6** (1992) 131-141.
4. V. Atalay and E. Gelenbe, "Parallel algorithm for colour texture generation using the random neural network model", *Int. J. of Pattern Recognition and Artificial Intelligence* **6** (1992) 437-446.
5. P. Baldi, "Neural networks, orientations of the hypercube, and algebraic threshold functions", *IEEE Trans. on Information Theory* **34** (1988) 523-530.
6. A. Dembo, "On the capacity of associative memories with linear threshold functions", *IEEE Trans. on Information Theory* **35** (1989) 709-720.
7. E. Gelenbe, "Random neural networks with negative and positive signals and product form solution", *Neural Computation* **1** (1989) 502-510.
8. E. Gelenbe, "Stability of the random neural network model", *Neural Computation* **2** (1990) 239-247.
9. E. Gelenbe, "Product form queueing networks with negative and positive customers", *J. Applied Probability* **28** (1991) 656-663.
10. E. Gelenbe, A. Stafylopatis and A. Likas, "Associative memory operation of the random network model", in *Artificial Neural Networks*, eds. T. Kohonen *et al.*, North-Holland, 1991, vol. 1, pp. 307-312.
11. E. Gelenbe, "G-networks with triggered customer movement", *J. Applied Probability* **30** (1993) 742-748.

12. E. Gelenbe, "Learning in the recurrent random neural network", *Neural Computation* 5 (1993) 154-164.
13. E. Gelenbe, "Hopfield energy of the random neural network", *Proc. IEEE Int. Conf. on Neural Networks*, Orlando, Florida, June 1994, vol. VII, pp. 4681-4686.
14. W. Henderson, "Queueing networks with negative customers and negative queue lengths", *J. Applied Probability* 30 (1993) 931-942.
15. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities", *Proc. of the National Academy of Sciences, USA*, vol. 79, 1982, pp. 2554-2558.
16. A. Likas and A. Stafylopatis, "An investigation of the analogy between the random network and the Hopfield network", *Proc. ISCIS VI*, North-Holland, 1991.
17. R. J. McEliece, E. C. Posner, E. R. Rodemich, and S. S. Venkatesh, "The capacity of the Hopfield associative memory", *IEEE Trans. on Information Theory* 33 (1987) 461-482.
18. M. Morita, "Associative memory with nonmonotone dynamics", *Neural Networks* 6 (1993) 115-126.
19. A. Stafylopatis and A. Likas, "Pictorial information retrieval using the random neural network", *IEEE Trans. on Software Engineering* 18 (1992) 590-600.
20. S. S. Venkatesh and D. Psaltis, "Linear and logarithmic capacities in associative neural networks", *IEEE Trans. on Information Theory* 35 (1989) 558-568.
21. T. Wang, "Improving recall in associative memories by dynamic threshold", *Neural Networks* 7 (1994) 1379-1385.
22. S. Yoshizawa, M. Morita, and S. Amari, "Capacity of associative memory using a nonmonotonic neuron model", *Neural Networks* 6 (1993) 167-176.

Received 24 November 1995; revised 10 May 1996.



**Aristidis Likas** received the Diploma degree in electrical and electronics engineering in 1990, and the Ph.D. degree in computer science in 1994 both from the National Technical University of Athens. He is currently with the

Department of Informatics, University of Ioannina, Greece. His research interests include neural networks, optimization techniques and parallel processing.



**Andreas Stafylopatis** received the Diploma degree in electrical and electronics engineering in 1979 from the National Technical University of Athens and the Docteur Ingenieur degree in computer science in 1982 from the Uni-

versity of Paris-Sud, Orsay, France. Since 1984 he has been with the Department of Electrical and Computer Engineering at the National Technical University of Athens, where he is currently a Professor. His research interests include parallel processing and computational intelligence.

