Complex Systems **6** (1992) 1–12

Can Spurious States Be Useful?

Eduardo R. Caianiello
Dipartimento di Fisica Teorica e S.M.S.A., Universita' di Salerno,
84081 Baronissi, Salerno, Italy

Michele Ceccarelli
Istituto di Ricerca sui Sistemi Informatici Paralleli, IRSIP-CNR
80125 Napoli, Italy

Maria Marinaro
Dipartimento di Fisica Teorica e S.M.S.A., Universita' di Salerno,
84081 Baronissi, Salerno, Italy

Abstract. Additive automata are analyzed and used as associative memories. Their storage capacity is computed as well as the attraction basin of each memorized pattern. We show how such systems can be used when the patterns to be memorized have a subset of relevant features; "spurious states" become meaningful as carriers of additional information, and storage capacity is thereby increased.

1. Introduction

A problem in the study of associative network models that deserves further investigation is how the storage capacity of the network depends on the structure of the patterns being memorized. Previous work (see for example [8]) on this subject, making use of a statistical approach, determined the dependence of the theoretical limit of the capacity on the correlation among the patterns, and proved that the storage capacity increases with the correlation. To our knowledge, the influence of other types of relationships among patterns on storage capacity has not yet been fully analyzed, though further knowledge would be very useful both from a theoretical point of view and for applications, as it might allow the construction of specific-purpose networks with large capacity.

In this paper we investigate this problem for the particular class of networks known as *additive automata* [6, 15]. We shall show that, preassigned p arbitrary patterns in $\{0, 1, \ldots, k-1\}^N$, $N \geq p$, it is possible to construct an additive automaton of N processing elements, "neurons," which stores k^p patterns; these patterns are not arbitrary, but are related among themselves by being linear combinations of the p preassigned patterns. Each of the k^p stable states has a well-defined basin of attraction whose cardinality is k^{N-p}, and each basin is simply a translation of a fixed linear subspace of the state space.

In section 2 we study the evolution of an additive automaton and show that all of its well-known properties can be determined by computing the rank of the synaptic matrix and of its powers. In section 3, by considering additive automata that act as associative memories, we compute their storage capacity and give a learning algorithm to construct the matrix of connections once p arbitrary patterns are preassigned. A simple example that utilizes spurious states for this purpose is given.

In the following we shall use the name "net" or "automaton" indifferently.

2. Description and properties of the model

Let us indicate the state of the net of N processing elements at time t with the N-component vector $\underset{\sim}{\xi}^t = (\xi_1^t, \xi_2^t, \ldots, \xi_N^t)$, where $\xi_i^t \in \{-1, +1\}$. The evolution law, by using a discretization of time, is

$$\underset{\sim}{\xi}^t = \sigma[\mathbf{F}(\underset{\sim}{\xi}^{t-\tau})] \qquad \text{or} \qquad \xi_i^t = \sigma[F_i(\underset{\sim}{\xi}^{t-\tau})] \tag{1}$$

where τ is the time unit, $\sigma[x] = +1$ $(x > 0)$ or -1 $(x < 0)$ for $x \neq 0$, and $\sigma[\underset{\sim}{x}]$ means that σ is applied to each component of the vector $\underset{\sim}{x}$. It is well known [2] that for any F_i we can write

$$\xi_i^t = \sum_{\alpha=0}^{2^N-1} f_\alpha^i \eta_\alpha^{t-\tau} \tag{2}$$

where the coefficients f_α^i are constants whose values are given in terms of the function $\mathbf{F}$, and η_α is a monomial of variables $\xi_j^{t-\tau}$, $j = 1, \ldots, N$. The sum over α covers all the distinct monomials of the N boolean variables $\xi_j^{t-\tau}$.

In this paper we shall be interested in systems for which the evolution law is additive with respect to the Hadamard product (the product componentwise, denoted by $*$) between two or more states, that is,

$$\sigma[\mathbf{F}(\underset{\sim}{\xi}_1 * \underset{\sim}{\xi}_2)] = \sigma[\mathbf{F}(\underset{\sim}{\xi}_1)] * \sigma[\mathbf{F}(\underset{\sim}{\xi}_2)].$$

In a previous work [3] it was shown that *in order to have such additive evolution it is necessary and sufficient that, for all i, the sum in (2) has one and only one coefficient not equal to 0 and equal to 1.*

So, we have that, for additive systems, the evolution law is a monomial:

$$\xi_i^t = \xi_{i_1}^{t-\tau} \cdot \xi_{i_2}^{t-\tau} \cdot \ldots \cdot \xi_{i_r}^{t-\tau}. \tag{3}$$

Because of the isomorphism between the two groups $(\{-1, +1\}, \cdot)$ and $(\{0, 1\}, +_2)$ where $+_2$ is summation modulo 2, equation (3) can be written as

$$x_i^t \equiv_2 \sum_{j=i_1, i_2, \ldots, i_r} x_j^{t-\tau} \tag{4}$$

with $x_i^t \in \{0, 1\}$, $i = 1, \ldots, N$.

The above considerations are based only on the additivity of systems, and can be generalized to the case in which the neuron can assume at each time one of k different states. In the following we study the generalized case and write the evolution law as

$$x_i^{t+\tau} \equiv_k \sum_j a_{i,j} x_j^t \tag{5}$$

with $x_j^t, a_{i,j} \in Z_k = \{0, 1, \ldots, k-1\}$ and k a prime. Equation (5) written in vectorial form is

$$\underline{x}^{t+\tau} \equiv_k A\underline{x}^t \qquad \underline{x}^t \in \{0, 1, \ldots, k-1\}^N \tag{6}$$

where A is an $N \times N$ matrix with elements in Z_k as in (5), and $\equiv_k$ means that the equality is taken modulo k. It is worth noting that the evolution law (6) is additive with respect to the sum of the states, that is, $\underline{z}^t \equiv_k \underline{x}^t + \underline{y}^t \Longrightarrow \underline{z}^{t+\tau} \equiv_k \underline{x}^{t+\tau} + \underline{y}^{t+\tau}$.

As a consequence of the simple evolution law (6) the additive automata verify the following properties:

1. The state $\underline{x}_0$ whose components are all equal to zero is stable:

   $$A\underline{x}_0 \equiv_k \underline{x}_0$$

2. The number of immediate predecessors of a given state, namely the states that in one step evolve into it, does not depend on the state. In fact, let N_0 be the number of states that evolve in one step in $\underline{x}_0$, and call H_0 the subspace containing all these states. It is evident that the N_0 states obtained as a sum of a state $\underline{x}$ of the automaton and one of the states in H_0

   $$\underline{z} \equiv_k \underline{x} + \underline{j}_0 \qquad \underline{j}_0 \in H_0$$

   go all into the same state $A\underline{x}$.

3. The main characteristics of the evolution or the number of transient states, the number of fixed states, the number of cycles and their lengths are all directly connected with the properties of the synaptic matrix A.

The constraint of additivity produces severe restrictions on the computations that can be performed by such systems; in fact, the prescription about the evolution of a set of states also fixes the evolution of their "linear" combinations.[1] To show this we observe that *the additive system is completely specified by fixing, a priori, the transition of N linearly independent states.* In fact, if M is an $N \times N$ matrix whose columns are N linearly independent states and M' contains the corresponding states at the next step we want a matrix A such that

$$AM \equiv_k M'.$$

[1] Here and in what follows we say that a vector $\underline{x} \in Z_k^N$ is a linear combination of vectors $\underline{x}_1, \ldots, \underline{x}_l \in Z_k^N$ if l constants $c_1, \ldots, c_l \in Z_k$, not all vanishing, exist such that $\underline{x} \equiv_k c_1\underline{x}_1 + c_2\underline{x}_2 + \ldots + c_l\underline{x}_l$.

Since M is non-singular

$$A \equiv_k M'M^{-1}$$

where M^{-1} is the inverse of M computed in algebra modulo k.

The knowledge of the matrix A allows us to determine the complete behavior of the additive automaton; first we prove the following.

Theorem 1. *If $r = rank(A)$ and m is the minimum integer such that $rank(A^m) = rank(A^{m+1})$, then*

(*a*) *the transient states disappear in at most m steps; and*

(*b*) *the number of states that survive is $k^{mr-(m-1)N}$*

Proof. First we observe that, since the columns in A^l are linear combinations of the columns in A^{l-1}, $\text{rank}(A^l) \leq \text{rank}(A^{l-1})$. We indicate with $r \geq r_1 \geq r_2 \geq \cdots$ the ranks of the matrices $A, A^2, A^3, \ldots$, respectively, and with ϕ the matrix containing all the k^N vectors in Z_k^N such that

$$\phi_{jh} = \left\lfloor \frac{h}{k^j} \right\rfloor \bmod k \qquad 0 \leq j \leq N-1,\ 0 \leq h \leq k^N - 1$$

where $\lfloor x \rfloor$ is the greatest integer equal to or less than x. Afterwards we construct the matrices

$$\phi_1 \equiv A\phi, \quad \phi_2 \equiv A^2\phi, \quad \phi_3 \equiv A^3\phi, \quad \ldots,$$

which contain a decreasing number of distinct states

$$k^r \geq k^{r_1} \geq k^{r_2} \geq \cdots$$

Since $r_i \geq 0$, a number m such that $r_{m-1} = r_m$ exists. At this point the application of A causes only a permutation of the distinct states of ϕ_{m-1}; we deduce that all these states are on cycles. This proves point (a), while to prove (b) we observe that as a consequence of property (2) each state in ϕ_1 has k^{N-r} immediate predecessors, and this is true also for the states in ϕ_2. Thus

$$k^{r_1} = \frac{k^r}{k^{N-r}} = k^{2r-N}$$

By applying the same considerations to $\phi_3, \ldots, \phi_m$ we obtain

$$r_m = r_{m-1} = mr - (m-1)N$$

Thus the states that survive are

$$k^{mr-(m-1)N}$$ ■

The number m indicates the maximum depth of the attraction basins, that is, the maximum number of steps after which the system reaches an attractor; since $mr - (m-1)N \geq 0$, it is bounded by $N/(N-r)$ for $r \neq N$, while it is zero for $r = N$.

As a consequence of the previous theorem we have the following corollary.

Corollary.

1. *The number of transient states is given by*

$$k^N - k^{mr-(m-1)N}$$

2. *The cardinality of the attraction basin of a stable state is*

$$k^{m(N-r)}$$

3. *The cardinality of the attraction basin of a length i cycle is given by*

$$ik^{m(N-r)}$$

Theorem 2. *The number of cycles of length l, denoted by n_l, is given by*

$$n_l = \frac{k^{N-v_l} - \sum_j k^{N-v_j}}{l}$$

where $v_l = \mathrm{rank}(A^l - I)$ and the sum is over all j that divide l.

Proof. If l is a prime and $\underline{x}$ belongs to a cycle of length l, we have $A^l\underline{x} \equiv \underline{x} \Longrightarrow (A^l - I)\underline{x} \equiv 0$. Then it is clear that the number of distinct states belonging to cycles of length l is k^{N-v_l} because the $\mathrm{Ker}(A - I)$ contains $N - v_l$ linearly independent states, and, as a conseqence of the additivity, all these states and their linear combinations belong to cycles of length l. Generalization to the case in which l is divisible is immediate. ∎

Finally we prove that the lengths of the cycles are not arbitrary, but have to satisfy the following condition.

Theorem 3. *The lengths of the cycles divide $q = \max\{i : n_i \neq 0, i \geq 1\}$.*

Proof. If $\underline{x}$ belongs to cycle of length q $(A^q\underline{x} \equiv \underline{x})$, and $\underline{y}$ belongs to a cycle of length v $(A^v\underline{y} \equiv \underline{y})$, we suppose *ab absurdo* that a $v < q$, $v \neq 1$ exists such that $n_v \neq 0$ and $\mathrm{MCD}(v, q) = 1$. Then the state $\underline{x} + \underline{y}$ will belong to a cycle of length vq, $A^{vq}(\underline{x} + \underline{y}) \equiv A^{vq}\underline{x} + A^{vq}\underline{y} \equiv \underline{x} + \underline{y}$, and $qv = \mathrm{lcm}(q, v)$ is the least length of the cycle of $\underline{x} + \underline{y}$. The fact that $vq > q$ proves the theorem. ∎

These results will be useful for analyzing in a straightforward manner the behavior of such systems for association tasks, as shown in the next section.

3. Associative memory

In this section we consider additive automata that act as associative memories and, for the sake of simplicity, we study Instantaneous Associative Memories (IAMs) [6, 7], in which the system, starting from any initial state, reaches a stable state in one step of the evolution. It is very simple to characterize an IAM: indeed as a consequence of Theorems 1 and 2 we have that an additive automaton is an IAM if and only if $r = N - v$, where $r = \text{rank}(A)$ and $v = \text{rank}(A - I)$. On the other hand, we can construct an IAM by using the results found in the previous section. In fact, if we want to memorize p arbitrary linearly independent states $\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_p$, we can build an $N \times N$ matrix φ whose columns are formed by the vectors $\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_p$ and by other $N-p$ arbitrary vectors $\underline{J}_1, \ldots, \underline{J}_{N-p}$ such that all N vectors form a basis of Z_k^N. A convenient notation for φ is the following:

$$\varphi = (\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_p, \underline{J}_1, \ldots, \underline{J}_{N-p}).$$

We want the next step evolution of the states to be given by the matrix φ',

$$\varphi' = (\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_p, \underline{x}_0, \underline{x}_0, \ldots, \underline{x}_0),$$

whose first p columns coincide with those of matrix φ while the other $N-p$ have all the elements equal to zero. The synaptic matrix A of this automaton is given by [10, 11, 12, 14]

$$A \equiv_k \varphi' \varphi^{-1}. \tag{7}$$

The automaton so constructed is obviously an IAM and has k^p stable states: the states $\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_p$ and all their linear combinations. The basin of attraction of each stable state has cardinality k^{N-p}; in fact, if we call H_J the subspace of Z_k^N spanned by $\underline{J}_1, \ldots, \underline{J}_{N-p}$, it is evident that the attraction basin of any other state $\underline{x}_n$ contains the k^{N-p} vectors

$$\underline{z}_i \equiv \underline{x}_n + \underline{y}_i \qquad \underline{y}_i \in H_J, n = 1, \ldots, p,$$

and the structure of the attraction basins is completely characterized by the choice of the vectors $\underline{J}_1, \ldots, \underline{J}_{N-p}$.

Until now we have shown that an additive net of N neurons can memorize a very large number of states, k^p $(p \leq N)$, which of course are not arbitrary, being combinations of p linearly independent states. Each of the stored states has a well-defined basin of attraction of cardinality k^{N-p}. The structure of the basin is determined by the k^{N-p} vectors that belong to the attraction basin of the state $\underline{x}_0$.

Relation (7) is characteristic of associative network models that make use of spectral algorithms [11, 12, 13, 15]. For such algorithms the matrix φ is not square, and φ^{-1} is replaced by the so-called generalized inverse. However, there are other methods to construct the matrix of connections that make use of outer-product algorithms [8, 14]. In the following we give two outer-product algorithms: the first memorizes one pattern at each application and maintains the properties of (7) (IAM); the second performs hetero-associations among states. Both algorithms start with an $N \times N$ matrix with all elements equal to zero, $A^{(0)} = \mathbf{0}_N$.

Algorithm 1. At the ith step, when $i-1$ states have been stored and we want to store another state $\underline{\xi}^{(i)}$, we put

$$A^{(i)} \equiv_k A^{(i-1)} + (A^{(i-1)}\underline{\xi}^{(i)} - \underline{\xi}^{(i)})\underline{\eta}^{(i)T} \tag{8}$$

where the second term of the right side is an outer product of two vectors and

(a) $A^{(i-1)}$ is the synaptic matrix that stores the $i-1$ previous states;

(b) $\underline{\eta}^{(i)} \in \{0,1,\ldots,k-1\}^N$ is a vector such that

$$\underline{\eta}^{(i)} \equiv_k z_j^{-1}[\underline{e}_j - \underline{r}_j(A^{(i-1)})]\,;$$

(c) $\underline{z} \equiv_k A^{(i-1)}\underline{\xi}^{(i)} - \underline{\xi}^{(i)}\,;$

(d) j is chosen such that $z_j \neq 0\,;$

(e) $\underline{e}_j$ has all the components equal to zero except the jth, which is equal to 1; and

(f) $\underline{r}_j^T(A^{(i-1)})$ is the jth row of $A^{(i-1)}$.

If $\underline{\xi}^{(i)}$ is already stable then $\underline{z}^{(i)} \equiv \underline{0}$.

Theorem 4. *The rule (8) allows the memorization of the new pattern $\underline{\xi}^{(i)}$ and preserves the previous ones.*

Proof. If we multiply $A^{(i)}$ by $\underline{\xi}^{(i)}$, according to (8), we obtain

$$A^{(i)}\underline{\xi}^{(i)} \equiv A^{(i-1)}\underline{\xi}^{(i)} + (A^{(i-1)}\underline{\xi}^{(i)} - \underline{\xi}^{(i)})\underline{\eta}^{(i)T}\underline{\xi}^{(i)}. \tag{9}$$

Since

$$\begin{aligned}\underline{\eta}^{(i)T}\underline{\xi}^{(i)} &\equiv z_j^{-1}[\underline{e}_j^T - \underline{r}_j^T(A^{(i-1)})]\underline{\xi}^{(i)} \\ &\equiv \left(\sum_k a_{j,k}^{(i-1)}\xi_k^{(i)} - \xi_j^{(i)}\right)^{-1}\left(\xi_j^{(i)} - \sum_k a_{j,k}^{(i-1)}\xi_k^{(i)}\right) \equiv -1,\end{aligned}$$

equation (9) becomes

$$A^{(i)}\underline{\xi}^{(i)} \equiv \underline{\xi}^{(i)}.$$

Furthermore, the patterns $\underline{\xi}^{(1)},\ldots,\underline{\xi}^{(i-1)}$ remain memorized because, for $\underline{\xi}^{(i-1)}$,

$$\underline{\eta}^{(i)T}\underline{\xi}^{(i-1)} = z_j^{-1}(\xi_j^{(i-1)} - \sum_k a_{j,k}^{(i-1)}\xi_k^{(i-1)}) = z_j^{-1}(\xi_j^{(i-1)} - \xi_j^{(i-1)}) = 0$$

and obviously

$$\begin{aligned}A^{(i)}\underline{\xi}^{(i-1)} &\equiv A^{(i-1)}\underline{\xi}^{(i-1)} + (A^{(i-1)}\underline{\xi}^{(i-1)} - \underline{\xi}^{(i-1)})\underline{\eta}^{(i)T}\underline{\xi}^{(i-1)} \\ &\equiv A^{(i-1)}\underline{\xi}^{(i-1)} \equiv \underline{\xi}^{(i-1)}.\end{aligned}$$

∎

The matrices generated by Algorithm 1 are very sparse, and each application of (8) adds only one column different from zero to the matrix of connections, so the space required to store a matrix that memorizes p linearly independent states of length N is $O(pN)$, and, as a consequence, there is no waste of space using this kind of system for associative memories. The *efficiency* of a network model can be defined as the ratio between the total number of bits stored in the net and the number of bits needed to store the synaptic matrix of the connections. For most network models studied thus far (see [1] for a review), the efficiency is generally very low and decreases with N. In our case the efficiency is maximal, $O(1)$, and there is still a distributed representation and a noise-correction capability as shown below.

Let $j_1, \ldots, j_p$ be the sequence of the choices of the index j at each application of the algorithm. This sequence gives a specific associative task to the system, that is, it determines the structure of the attraction basins. It corresponds to the choice of the vectors $\underline{J}_1, \underline{J}_2, \ldots, \underline{J}_{N-p}$ of the matrix φ in (7), with the only constraint being that such vectors have only one component different from zero and equal to one. If arbitrary vectors $\underline{J}_1, \ldots, \underline{J}_{N-p}$ are used, then the following algorithm can be applied.

Algorithm 2. The hetero-association among the states in Z_k^N, $\underline{\xi}^{(i)} \rightarrow \underline{\eta}^{(i)}$ can be performed as follows:[2]

$$A^{(i)} \equiv_k A^{(i-1)} + (A^{(i-1)}\underline{\xi}^{(i)} - \underline{\eta}^{(i)})\underline{\rho}^{(i)T} \tag{10}$$

where $\underline{\rho}^{(i)T}$ is a vector with elements belonging to Z_k^N, such that the following holds:

$$\begin{aligned} &\underline{\xi}^{(j)T}\underline{\rho}^{(i)} \equiv_k 0 \qquad j = 1, \ldots, i-1 \\ &\underline{\xi}^{(i)T}\underline{\rho}^{(i)} \equiv_k -1 \end{aligned} \tag{11}$$

Also in this case there is a choice that characterizes the structure of the evolution because many vectors $\underline{\rho}^{(i)}$ satisfying the condition in (11) can exist. In this way we could classify the algorithms as "unsupervised" learning if the choice is random, and as "supervised" learning if the choice is guided from the outside.

Theorem 5. *The rule* (10) *memorizes the association* $\underline{\xi}^{(i)} \rightarrow \underline{\eta}^{(i)}$ *and preserves the previous ones.*

Proof. The proof is the same as that of Theorem 4:

$$A^{(i)}\underline{\xi}^{(i)} \equiv A^{(i-1)}\underline{\xi}^{(i)} + (A^{(i-1)}\underline{\xi}^{(i)} - \underline{\eta}^{(i)})\underline{\rho}^{(i)T}\underline{\xi}^{(i)} \equiv \underline{\eta}^{(i)}$$

The proof that the previous associations are preserved is trivial. ■

[2] Note that $\underline{\xi}^{(i)}$ could be equal to $\underline{\eta}^{(i)}$.

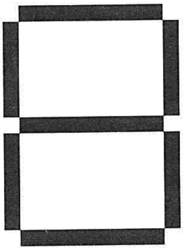

Figure 1: The memorized patterns are the seven bars constituting the figure.

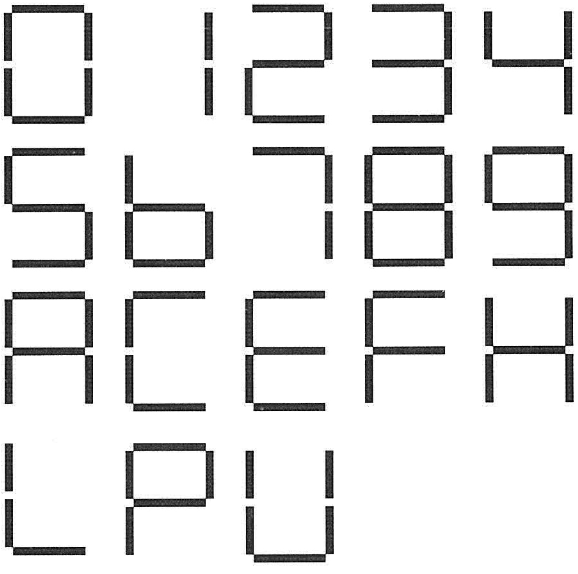

Figure 2: Some attractors of the network.

As an example we memorize seven binary patterns of dimensions 16 × 16 pixels each representing a bar of figure 1. The memorization is carried out with Algorithm 1 and a random choice of the index j at each step. The system so constructed has 128 attractors, which are the combinations of the seven patterns given. Here a novel situation arises, which justifies the title of this paper: it seems that "spurious states" may turn out to be an asset instead of a liability. Indeed, a number of combinations of the seven basic patterns memorized (which are customarily considered "spurious" and therefore unwanted) turn out to be meaningful and thereby useful. For instance, we find in figure 2 eighteen meaningful patterns that come out in addition to those originally stored. Furthermore, the same result is obtained

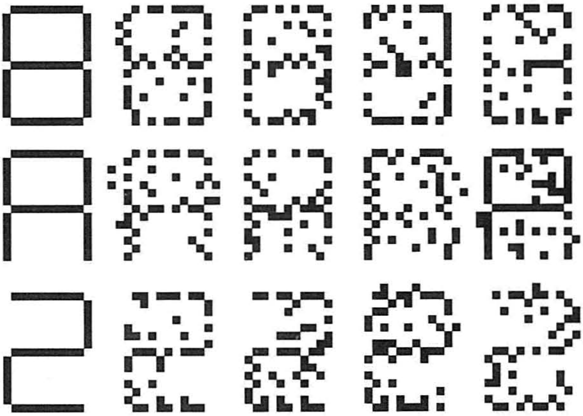

Figure 3: Noisy patterns well classified by the network. All the patterns in each row are attracted, after just one step of the evolution, by the pattern in the first column.

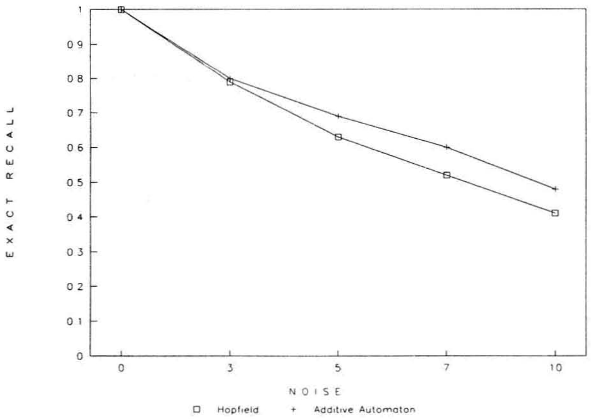

Figure 4: Exact recall vs. noise percent rate.

if we start from any seven independent combinations of the original patterns, for example "0," "1," "2," "3," "4," "5," and "7."

To appreciate this fact we may consider what happens in this case with the standard Hopfield model [13]. If we start with the original set of (orthogonal) patterns we retrieve the same situation as with our algorithm; if we start instead with (not orthogonal) combinations of them, we lose convergence even to the primitive set.

Finally, our procedure shows a remarkable noise correction ability: figure 3 shows how some quite noisy patterns are exactly recognized in one step. To evidence the noise correction ability we plot in figure 4 the exact recall probability as a function of the noise percentage both for the Hopfield model and for the additive network constructed with Algorithm 1 for the same set of patterns. We used as a training set the seven bars constituting figure 1. A test set was generated by constructing 200 noisy vectors for each pattern of figure 2 and for each noise percentage. The recalling ability of the additive automaton compares favorably with that of the Hopfield model with the advantage of maximal efficiency and instantaneous recalling.

4. Conclusions

Spurious states as syllables

If we think of the basic seven patterns as "letters" of an alphabet, the 128 that are memorized in our example correspond to combinational strings of the letters. In any language, only a small number of such strings form meaningful syllables. As such we can take the $7 + 18$ of our example. This point may deserve further investigation.

Classification by features

As pointed out in the introduction, the systems described in the previous sections present a particular associative behavior. In our case, by fixing p states in memory, each attraction basin contains k^{N-p} vectors; these vectors have the characteristic that, in the p positions used to store the patterns (the choice of j in the Algorithm 1), they assume the same value. Thus, the attraction basin can be represented, in the binary case, as a set of vectors in this way:

$$(\ldots, *, \ldots, 1, \ldots, *, \ldots, 0, \ldots)$$

where $*$ can take any of the values $0, 1$ and only p positions are fixed. In this way an attraction basin can capture an assigned characteristic of the pattern, namely a given subset of components of the pattern that is relevant. Thus, these models are a powerful associative mechanism when we want to learn patterns on the basis of their features, rather than classing them according to some distance, such as the Hamming distance.

Acknowledgments

This work was supported in part by CNR Progetto Strategico "Reti Neurali," by INFM, and by I.I.A.S.S.

References

[1] L. F. Abbott, "Learning in Neural Network Models," *Network*, **1** (1990) 105–122.

[2] E. R. Caianiello, "Outline of a Theory of Thought-Processes and Thinking Machines," *Journal of Theoretical Biology*, **1** (1961) 204–231.

[3] E. R. Caianiello, "Neuronic Equations Revisited and Completely Solved," in *Brain Theory*, edited by G. Palm and A. Aertsen (Berlin, Springer Verlag, 1986).

[4] E. R. Caianiello and M. Marinaro, "Linearization and Syntheses of Cellular Automata: The Additive Case," *Physica Scripta*, **34** (1986) 444–448.

[5] M. Ceccarelli and A. Petrosino, "Convergence and Processing in Learning for Neural Nets: The AAM Model," in *Parallel Architecture and Neural Networks*, second Italian workshop, edited by E. R. Caianiello (Singapore, World Scientific, 1990).

[6] M. Ceccarelli, A. Petrosino, and R. Tagliaferri, "Additive Automata and Associative Memories," in *Proceedings of the IJCNN-90, Washington, D.C.*, edited by M. Claudill (Hillsdale, NJ, Lawrence Erlbaum Associates, 1990).

[7] M. Ceccarelli, A. Petrosino, and R. Tagliaferri, "Dynamics and Associative Mapping in Additive Systems," *Proceedings of the ICNN-90, Paris* (Norwell, MA, Kluwer Academic Publishers, 1990).

[8] E. Gardner, "The Space of Interactions in Neural Network Models," *Journal of Physics A*, **21** (1988) 257–270.

[9] P. Guan and Y. He, "Exact Results for Deterministic Cellular Automata," *Journal of Statistical Physics*, **43** (1986) 463–478.

[10] I. Kanter and H. Sompolinsky, "Associative Recall of Memories without Errors," *Physical Review A*, **35** (1987) 380–385.

[11] T. Kohonen, *Self-Organization and Associative Memory* (Berlin, Springer Verlag, 1984).

[12] L. Personnaz, I. Guyon, and G. Dreyfus, "Information Storage and Retrieval in Spin-Glass Like Neural Networks," *Journal de Physique*, **46** (1985) 359–365.

[13] J. J. Hopfield, "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," *Proceedings of the National Academy of Sciences*, **79** (1982) 281–290.

[14] S. S. Venkatesh and D. Psaltis, "Linear and Logarithmic Capacities in Associative Neural Networks," *IEEE Transactions on Information Theory*, **35** (1989) 558–568.

[15] S. Wolfram, *Theory and Applications of Cellular Automata* (Singapore, World Scientific, 1986).