Collective computational properties of neural networks: New learning mechanisms

L. Personnaz, I. Guyon, and G. Dreyfus
Laboratoire d'Electronique, Ecole Supérieure de Physique et de Chimie Industrielles de la Ville de Paris, 10 rue Vauquelin, 75005 Paris Cédex 05, France
(Received 2 April 1986)

We present two learning mechanisms for networks of formal neurons analogous to Ising spin systems. The "projection rule" guarantees the errorless storage and retrieval of a set of information patterns: In other words, it allows us to compute the magnetic interactions so as to make a given set of states the ground states of the spin system (in zero external field). Several analytical results are derived for this rule; computer simulations and examples of applications to error correction are presented. Another learning mechanism, termed the "associating rule," is also described; going beyond the memorization process, it allows us to design networks satisfying a set of dynamical constraints such as a given set of stable states and/or transitions and/or cycles. It provides a new tool to perform such functions as associations between information and concepts.

INTRODUCTION

The quest for parallel computing devices as alternatives to von Neumann computers has long been a very active field of research. The first attempt to use neuronlike structures in this context was due to Rosenblatt[1] (perceptrons); these investigations came to a halt with the realization that such systems had limited computing capabilities.[2] This subject gained a renewed interest when it was shown that nonlinear feedback, achieved in fully connected networks and absent in perceptrons, was an essential ingredient for performing high-order functions, and that the tools of statistical physics could be used efficiently to understand the behavior of such highly complex systems. A first decisive step towards a more comprehensive understanding occurred when Little[3] recognized the temperature-noise analogy. Another important element was brought by Hopfield,[4] who showed that fully connected networks of formal neurons exhibited computational and information-processing abilities. From this standpoint, the behavior of a network can be viewed as a decision-making process which is distributed both in space (each neuron takes its own decision) and in time (the decision is taken in one or more steps). This is in sharp contrast to one-step information processing systems such as, for instance, linear associative memories or "perceptrons."

During the last four years, a growing number of investigations were inspired by the analogy between spin systems and neural networks. The statistical mechanical properties of the Hopfield networks have been analyzed in detail[5–7] and several phase transitions between paramagnetic, ferromagnetic, and spin-glass phases have been predicted. Pattern-recognition properties have been discussed[8] and a modified model, involving a non-Hebbian learning rule, has been proposed;[9] the finite-temperature behavior of the latter model, and the resulting phase diagram, have also been analyzed.[7,10] Models of learning and information encoding, which might be relevant to the biological nature of memory, have been proposed recently.[11–13]

In the present paper we investigate the behavior of neural networks designed with a learning rule (projection rule) which, in contrast to the classical Hebb's rule, guarantees the memorization and retrieval of a given information: This rule allows one to dig holes in the energy landscape at predetermined points; we explain the "decision" mechanism of one parallel iteration of the network, we examine the nature of the stable states of the system, we determine the minimum size of the basins of attraction of the memorized states, and we show that no cycles can occur. Moreover, this rule can easily be generalized; we prove that it is possible to design networks which comply with various sets of transition constraints in state space, as, for instance, transient sequences leading to an attractor, cycles, etc. Therefore, such networks have the ability of performing associations between informations (associating rule). These rules are illustrated by various examples of automatic error correction and classification.

I. PRESENTATION OF THE NETWORK

A. The network of formal neurons

The formal neurons investigated in this work are deterministic threshold elements with several binary inputs and one binary output. The state of any such neuron i, which is its output value, is represented by a variable σ_i, the value of which can be either 1, if the neuron is "active," or -1 if the neuron is "inactive."

We consider fully connected networks of n such neurons operating in parallel with period τ, without sensory inputs. Therefore, the formal neuron i has n binary inputs σ_j, $j=1,2,\ldots,n$, which are the outputs of the n neurons of the network; the strength of the "synaptic" junction of neuron i receiving information from neuron j is represented by a coupling coefficient C_{ij} (without any *a priori* constraints such as $C_{ii}=0$ and/or $C_{ij}=C_{ji}$).

The state of the formal neuron i at time $t+\tau$ depends on the states of its inputs at time t in the following way:

the neuron first computes its "membrane potential" $v_i(t)$ by performing the sum of its inputs $\sigma_j(t)$ weighted by the coupling coefficients C_{ij},

$$v_i(t)=\sum_{j=1}^{n} C_{ij}\sigma_j(t) .$$

Then it compares $v_i(t)$ to a threshold value θ_i and determines its next state $\sigma_i(t+\tau)$ according to the following decision rule:

$$\begin{aligned} &v_i(t)-\theta_i\neq 0 \Longrightarrow \sigma_i(t+\tau)=\mathrm{sgn}[v_i(t)-\theta_i] ,\\ &v_i(t)-\theta_i=0 \Longrightarrow \sigma_i(t+\tau)=\sigma_i(t) . \end{aligned} \quad (1)$$

The n^2 coupling coefficients C_{ij} can be organized in a square matrix C, which is often called the "synaptic" matrix. The n threshold values define a column vector $\boldsymbol{\theta}$. At time t, the state of the network will be represented by a column vector $\boldsymbol{\sigma}(t)$, the n components of which are the states of the n neurons. The next state $\boldsymbol{\sigma}(t+\tau)$ of the network is fully determined by the computation of the vector $\mathbf{v}(t)$, the components of which are the potentials of the n neurons: $\mathbf{v}(t)=C\boldsymbol{\sigma}(t)$, followed by the decision rule (1); vector $\boldsymbol{\sigma}(t+\tau)$ is the vector $\mathbf{v}(t)$ "thresholded."

B. Associative properties of the network

We are interested in the computational properties of the networks, which arise from their spontaneous evolution: a network being set into an initial state, it will evolve step by step after decision rule (1); since the number of states is finite, and since rule (1) is deterministic, the evolution will end up in a limit cycle with a maximum length of 2^n. Several cases can arise; they are illustrated in Fig. 1, in which various possible evolutions in state space are represented: attractor or nonattractor fixed points and cycles. Obviously, a given network exhibits several fixed points or cycles, thus performing a partition of state space.

The case of Fig. 1(a) is certainly the most interesting since it exhibits typical associative properties. The state of a network of n neurons at a given instant of time can be considered as a pattern of n bits of information. Assume that the pattern corresponding to the fixed point 30 is an information which has been memorized, and that state 23 is an unknown pattern; if the network is set in state 23 it will evolve, by its own dynamics, to state 30, thus performing an association between informations 23 and 30. From a practical standpoint, it is much easier to detect that a system has reached a fixed point than to detect that it has entered a cycle.

Note that, since we consider parallel iterations, the fixed points (time-invariant states) are not necessarily stable states since the term "stability" refers strictly to single spin flips; therefore, the stability *stricto sensu* is not relevant in the present case. However, in the following, we shall refer to stable states in a somewhat loose way, as equivalent to fixed points.

The dynamics of the network represented by the above graph of state space depends on the parameters of the system, namely, the coupling coefficients and the thresholds. Therefore, the problem of designing an associative memory with a neural network can be stated as follows: how to determine the coupling coefficients and the thresholds so as to impose on the system a given set of attractors, which are the states to be memorized, or, more generally, so as to impose a set of dynamic behavior constraints. This problem will be addressed in the following.

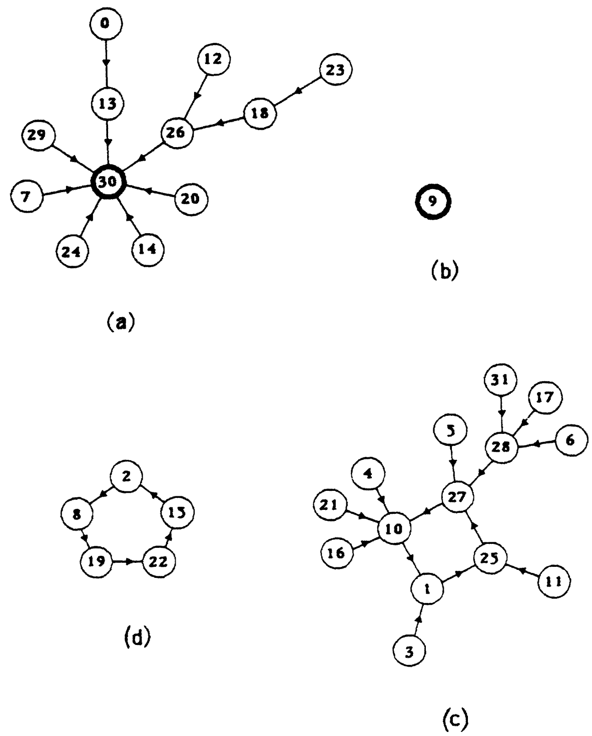


FIG. 1. Evolution in state space. A state $\boldsymbol{\sigma}$ of the network is represented by a labeled circle. An arrow represents a parallel iteration. Stable states are characterized by heavy circles. (a) Stable attractor state. (b) Stable nonattractor state. (c) Attractor cycle. (d) Nonattractor cycle.

II. A LEARNING RULE FOR RELIABLE INFORMATION RETRIEVAL

A. Presentation of the problem

In this section, we address the problem of designing a neural network which, by a proper choice of the parameters, actually memorizes a given set of prototype vectors by making them stable states of the network.

We shall see that such a network exhibits information-retrieval properties: given a distorted version of a memorized prototype vector as the initial state of the network, the latter will generally converge to the prototype vector, thus retrieving the complete correct information. But we shall further see that, in addition to the prototype states, other stable states arise too. These nonprototype stable states can be useful or undesirable, depending on the application which is considered. If the initial state is too confused with respect to the prototype states, the network converges to one of the nonprototype stable states, thereby indicating that it is not able to "recognize" anything. In some cases, however, if the basin of attraction of such states is too large, they can introduce undesirable decisions. Therefore, it is important to be able to predict them, possibly in order to suppress them.

We can summarize the desirable features for a network to perform information retrieval efficiently: (i) the prototype states, which are to be memorized, should be stable and act as attractors, (ii) the nonprototype stable states, if any, should be predictable, (iii) no cycles should be allowed to occur. In Sec. II C we show that these three conditions can be met if the parameters of the network are determined by an appropriate method.

B. Hebb's learning rule

Experimental data on biological systems have led Hebb to propose a learning mechanism[14] whereby the synaptic coupling between two neurons is enhanced if both neurons are active at the same time; the simplest rule for computing the coupling coefficients following the above idea has been proposed by Cooper *et al.*,[15]

$$C_{ij}=(1/n)\sum_{k=1}^{p}\sigma_i^k\sigma_j^k,\quad \sigma_i^k=\pm 1 \tag{2}$$

where n is the number of neurons and p is the number of prototype states to be memorized $\{\boldsymbol{\sigma}^k\}$.

Numerical simulations[4] and theoretical considerations[6] have shown that, in most cases, this rule does not meet the first of the three conditions which we have shown to be mandatory for a neural network to operate properly. Therefore, it is not satisfactory as far as information retrieval properties are concerned. In Sec. II C we shall derive a generalization of Hebb's rule which allows the above three conditions to be met.

Relation (2) can be written in matrix form as

$$C=(1/n)\sum_{k=1}^{p}\boldsymbol{\sigma}^k(\boldsymbol{\sigma}^k)^T=(1/n)\Sigma\Sigma^T , \tag{3}$$

where $(\boldsymbol{\sigma}^k)^T$ denotes the transpose of vector $\boldsymbol{\sigma}^k$, Σ is the (n,p) matrix whose columns are the vectors $\boldsymbol{\sigma}^k$,

$$\Sigma=(\boldsymbol{\sigma}^1,\boldsymbol{\sigma}^2,\ldots,\boldsymbol{\sigma}^p) ,$$

and Σ^T is the transpose of matrix Σ.

In the very particular case in which the prototype states are mutually orthogonal, the coupling matrix C, computed from relation (3), is the orthogonal projection matrix (in Euclidean space) into the subspace spanned by the prototype vectors. This particular property has an important impact on the dynamical behavior of the network which is determined by the components of vector $C\boldsymbol{\sigma}-\boldsymbol{\theta}$: if each component of this vector is zero or has the same sign as the corresponding component of $\boldsymbol{\sigma}$, the state $\boldsymbol{\sigma}$ is stable. In the present particular case (orthogonal prototype states) each vector $\boldsymbol{\sigma}^k$ is invariant in the projection operation, so that one has

$$C\boldsymbol{\sigma}^k=\boldsymbol{\sigma}^k \text{ for all } k . \tag{4}$$

Since $\boldsymbol{\sigma}^k$ belongs to $\{-1,+1\}^n$, it follows from relation (1) that, if one has

$$-1\le\theta_i\le 1,\quad \text{for all } i$$

all the prototype states are certainly stable states of the system; this property guarantees a perfect retrieval of the stored information.

If the prototype states are not orthogonal, the above property (4) is no longer true. The stability of the prototype states could be obtained by a proper choice of the threshold vector $\boldsymbol{\theta}$; however, the determination of such a vector requires solving a system of $n\times p$ inequalities,

$$(C\boldsymbol{\sigma}^k-\boldsymbol{\theta})_i\sigma_i^k\ge 0,\quad 1\le i\le n,\ 1\le k\le p ,$$

whose unknowns are the n components θ_i of vector $\boldsymbol{\theta}$. This system cannot be solved easily and does not always have a solution.

To summarize, Hebb's rule is suitable for the design of an associative memory guaranteeing a perfect retrieval of the stored information if the prototype vectors are orthogonal. The orthogonality condition can be met in several situations such as, for instance, (i) if n is finite, the columns of a Hadamard matrix[16] provide a complete set of orthogonal vectors, the components of which are $+1$ or -1; (ii) in the thermodynamic limit ($n\to\infty$) with p finite, if the prototype vectors are taken randomly (components $+1$ or -1 with probability $\frac{1}{2}$); this case has been studied in detail by other authors.[3–6]

For all practical purposes, however, one has to deal with finite systems, and the informations to be stored are neither random in nature nor orthogonal; the prototype states will, in general, be correlated, causing a low storage capacity. Therefore, it is natural to attempt to find a coupling matrix exhibiting the same basic property [relation (4)] for any set of vectors.[9]

C. The projection learning rule

In this section we show that there exists a coupling matrix which guarantees the stability of a set of prototype vectors, whether correlated or not, and we analyze to what extent neural networks designed with such matrices can be useful as associative memories.

A nontrivial solution of the system of equations (4),

$$C\boldsymbol{\sigma}^k=\boldsymbol{\sigma}^k,\quad \text{for all } k$$

which can be written equivalently as

$$C\Sigma=\Sigma , \tag{5}$$

will be the orthogonal projection matrix into the subspace spanned by the prototype vectors family $\{\boldsymbol{\sigma}^k\}$,

$$C=\Sigma\Sigma^I , \tag{6}$$

where Σ^I is the Moore-Penrose pseudoinverse[17] of Σ.

Therefore, it will guarantee the stability of the prototype states if one has

$$-1\le\theta_i\le 1 . \tag{7}$$

Relation (6) will be termed the *projection rule.* The coupling matrix C, being an orthogonal projection matrix, is symmetrical. In Sec. III, we shall investigate a still more general case, in which the coupling matrix is not symmetrical.

One should note that apparently with the projection rule there is no storage-capacity limit, up to the total number of possible states of the system. Nevertheless, this property does not mean that the network will always achieve the desired associative memory function. As a matter of fact, the memory capacity can be expressed directly in terms of the rank r of the family of the p prototype vectors. If $r=n$ $(p \geq n)$, the projection matrix is the identity matrix and the 2^n states of the network are stable; the memory is degenerate. If $r<n$, the associative memory function is possible; the retrieval efficiency of a prototype (attractivity) will fall sharply as r/n becomes of the order of 0.5.

Therefore, it is possible to memorize more than n prototypes without complete memory degeneracy, the only condition being $r<n$. Among the p prototypes, p-r are therefore linear combinations of r linearly independent prototypes.

In the general case, the coupling matrix C can be computed conveniently, without matrix inversion, by an iterative algorithm.[17] It yields the exact solution of system (5) after a finite number of iterations, which is equal to the number of prototype vectors. This kind of computation is typical of a learning process: once the synaptic matrix has been computed from a given set of prototype vectors, the addition of one extra item of knowledge does not require that the whole computation be performed again; one just has to run one iteration, starting from the previous matrix. Therefore, memorization through the projection rule retains the same iterative nature as the classical Hebb's rule.

The following three points should be mentioned.

(i) In the particular case where the prototype vectors $\boldsymbol{\sigma}^k$ are linearly independent, the synaptic matrix C takes the form

$$C=\Sigma(\Sigma^T\Sigma)^{-1}\Sigma^T . \tag{8}$$

If the prototype vectors are orthogonal, the projection rule reduces exactly to the classical Hebb's rule (3) since one has

$$(\Sigma^T\Sigma)^{-1}=(1/n)I ,$$

where I is the identity matrix.

(ii) In analogy to magnetic systems, zero diagonal matrices have been used by several authors.[3–8,10] Since the diagonal coefficients of the projection matrix are smaller than or equal to one, the stability of the prototype states after canceling the diagonal terms is preserved, but their attractivity is altered.

(iii) Finally, one can guarantee the stability of the prototype states with the projection rule without any restriction on the thresholds: the scaling of the thresholds is directly related to the scaling of the matrix. If one has $-\lambda \leq \theta_i \leq +\lambda$ for all i one can just take $C=\lambda\Sigma\Sigma^I$.

D. Properties of the networks designed with the projection learning rule

In the following, we give a simple geometrical interpretation of the dynamics of the network, which allows us to establish several results: first, we derive the nature of the nonprototype stable states; second, we show that it is possible to define an energy which is always decreasing during the free evolution of the network, so that no cycles can occur; next, we show that the prototype states are the states of lowest possible energy; finally, we derive the size of the basins of attraction of the prototype states if the latter are orthogonal.

The thresholds are taken equal to zero (or, equivalently, in terms of magnetic systems, the external field is zero); this choice is natural since it leaves the largest possible stability margin.

1. Geometrical interpretation of the dynamics of the network; nature of the stable states

Let us summarize the steps occurring during a single parallel iteration of duration τ: the potential vector $\mathbf{v}(t)=C\boldsymbol{\sigma}(t)$ is evaluated; subsequently, the next state $\boldsymbol{\sigma}(t+\tau)$ is obtained by the thresholding operation (1).

In the following, we denote $\boldsymbol{\sigma}(t)$ by $\boldsymbol{\sigma}$ and $\boldsymbol{\sigma}(t+\tau)$ by $\boldsymbol{\sigma}'$. A geometrical interpretation is given on Fig. 2(a).

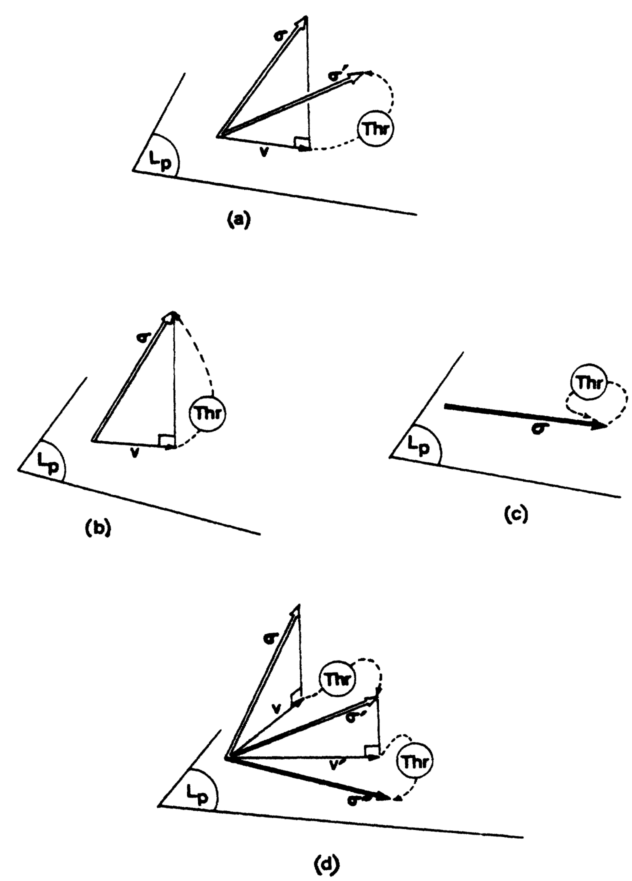
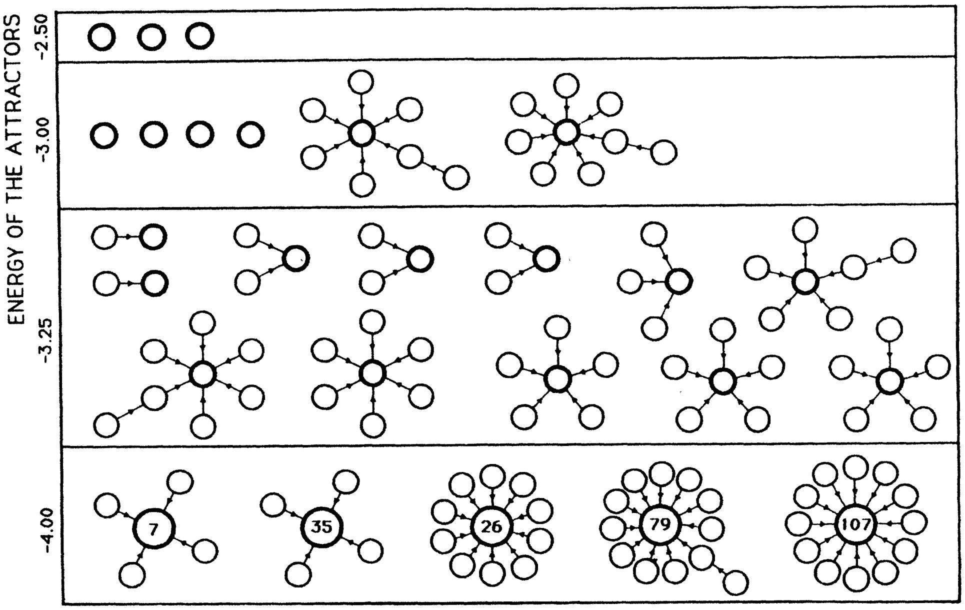
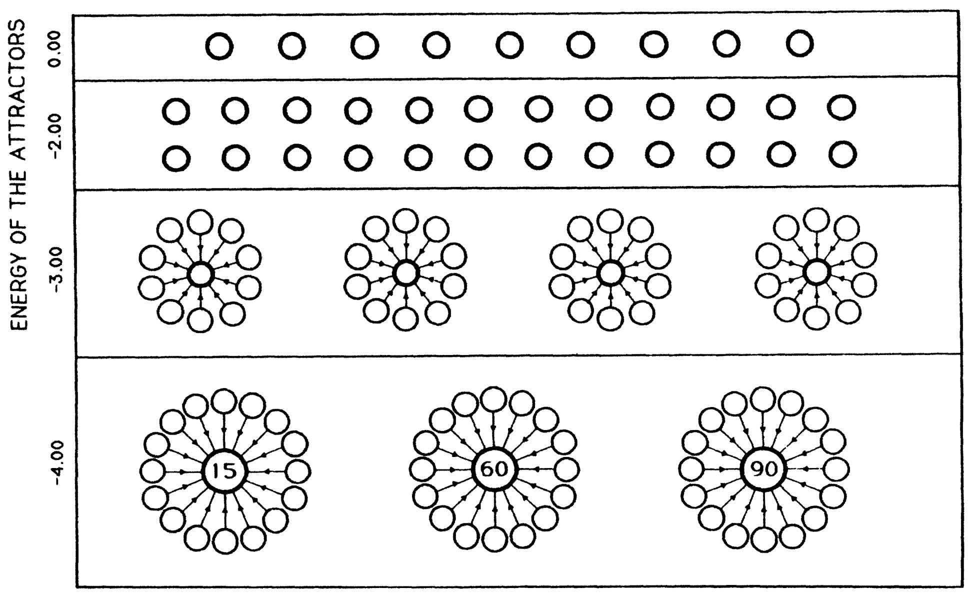


FIG. 2. Projection rule: Evolution in Euclidean space. (a) Evolution from state $\boldsymbol{\sigma}$ to state $\boldsymbol{\sigma}'$ in one parallel iteration. $\mathbf{v}=C\boldsymbol{\sigma}$ is the orthogonal projection of $\boldsymbol{\sigma}$ into the subspace L_p spanned by the p prototype vectors. $\boldsymbol{\sigma}'$, obtained after thresholding, is the vector belonging to $\{-1,+1\}^n$ which is closest to $\mathbf{v}$. (b) The vector $\boldsymbol{\sigma}$ is stable but does not belong to L_p. It is certainly not a prototype. (c) The vector $\boldsymbol{\sigma}$ is stable and belongs to L_p ($C\boldsymbol{\sigma}=\boldsymbol{\sigma}$; $\boldsymbol{\sigma}$ is, for instance, a prototype). (d) After two iterations from state $\boldsymbol{\sigma}$, the network reaches a stable state belonging to L_p.



(a)



(b)

FIG. 3. Projection rule: Graphs of evolution in state space. The labeled circles are the prototypes states: the label is the decimal value of the vectors considered as binary numbers (+ 1 is 1 and −1 is 0). The vertical scale shows the energy of the attractors. (a) Five linearly dependent prototype vectors are memorized. Matrix Σ^T,

−1	1	1	−1	1	−1	1	1	(107)
−1	−1	−1	1	1	−1	1	−1	(26)
−1	−1	−1	−1	−1	1	1	1	(7)
−1	−1	1	−1	−1	−1	1	1	(35)
1	−1	−1	1	1	1	1	1	(79) .

The rank of the family of the prototype vectors is 4. (b) Three orthogonal prototype vectors are memorized. Matrix Σ^T,

−1	−1	−1	−1	1	1	1	1	(15)
−1	1	−1	1	1	−1	1	−1	(90)
−1	−1	1	1	1	1	−1	−1	(60) .

Vector $\mathbf{v}=C\sigma$ is the orthogonal projection of σ into the subspace L_p spanned by the prototype vectors; therefore, $\mathbf{v}$ is the linear combination of the prototype states which is closest to the initial vector σ (Euclidean distance). We show in Appendix A that σ' (the thresholded vector $\mathbf{v}$) is the vector belonging to $\{-1,+1\}^n$ which is closest to $\mathbf{v}$. Consequently, if σ itself is the vector of $\{-1,+1\}^n$ which is closest to $\mathbf{v}$, σ is stable [Fig. 2(b)]; Fig. 2(c) illustrates a particular case where σ belongs to L_p (for instance, σ is a prototype). Figure 2(d) shows a sequence of iterations ending in a prototype state.

An immediate consequence of the above interpretation is the following: since $\mathbf{v}$ is a linear combination of the prototype states and σ' the thresholded vector $\mathbf{v}$, *the stable states are thresholded linear combinations of the prototype states.* If undesirable stable states appear, they may be eliminated according to the method suggested in Sec. III A 3.

2. *Dynamics of the network: absence of cycles, energy of the prototype states*

For further studies of the dynamical properties of the network, we use the following Lyapunov function of state σ:

$$E(\sigma)=-\tfrac{1}{2}(\sigma^T C\sigma)\ .$$

It should be noticed that, in contrast to the usual situation in investigations of magnetic systems or of neural networks, the diagonal terms of the coupling matrix being nonzero, this Lyapunov function is not strictly an energy function. Since C is an orthogonal projection matrix, one has

$$E(\sigma)=-\tfrac{1}{2}\mathbf{v}^T\mathbf{v}=-\tfrac{1}{2}\|\mathbf{v}\|^2\ ;$$

the "energy" is proportional to the square of the synaptic potential vector (or of the magnetization). We show in Appendix B that if a network evolves from a state σ to a state $\sigma'\neq\sigma$, one has $E(\sigma')<E(\sigma)$.

Thus, during the free evolution of the system performing parallel iterations, the energy is an ever decreasing function. Therefore, no cycles can occur; a similar result has been derived by other authors[4,18] for the evolution of a network in which one neuron only reevaluates its state at each time step (sequential operation); for practical purposes, however, parallel operation is more efficient as far as computation times are concerned.

The energy of a prototype state σ^k and its negative $-\sigma^k$ is given by

$$E=-\tfrac{1}{2}\sigma^{kT}C\sigma^k=-\tfrac{1}{2}\sigma^{kT}\sigma^k\ .$$

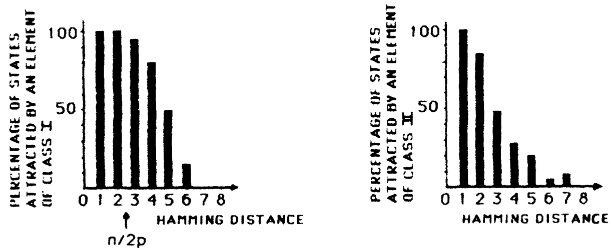
Thus, $E=-n/2$. Therefore, all the prototype states have the same energy, which is the lowest possible energy.

3. *Attractivity of orthogonal prototype states*

If the prototype states are orthogonal, their attractivity can be evaluated: we show in Appendix C that any state lying within a Hamming distance of $n/2p$ from a prototype state will converge to this state in one step. Therefore, the minimum number of states attracted by a given prototype state is given by

CLASSIFICATION OF THE ATTRACTORS

CLASS	I	II	III
NUMBER OF ELEMENTS	6	8	432
NUMBER OF ATTRACTED STATES	5553	2883	17
ENERGY OF AN ELEMENT	-8	-6	-4

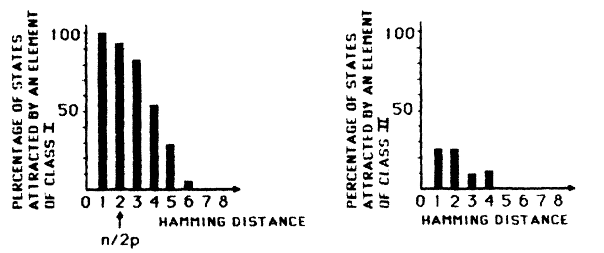


CLASS I (PROTOTYPES) CLASS II (SYMMETRIC STATES)

(a)

CLASSIFICATION OF THE ATTRACTORS

CLASS	I	II	III	IV	V	VI	VII
NUMBER OF ELEMENTS	8	32	64	384	128	384	432
NUMBER OF ATTRACTED STATES	3285	367	85	24	20	15	9
ENERGY OF AN ELEMENT	-8	-6	-6	-5	-4.5	-5.5	-4



CLASS I (PROTOTYPES) CLASS II (SYMMETRIC STATES)

(b)

FIG. 4. Projection rule: Histograms of attractivity. Network of 16 neurons. The orthogonal prototype vectors are from the columns of the Hadamard matrix. The stable states are gathered in classes of equal attractivity. Histograms of attractivity are presented for the first attractors. Class I: states belonging to L_p; prototypes, their negatives and possibly states which are linear combinations of prototypes. Class II: second attractors which are symmetric states. Class III: other attractors. (a) Matrix Σ,

(3855) (13107) (21845) .

(b) Matrix Σ,

(3855) (13107) (21845) (39321) .

$$\sum_{k=0}^{m} C_n^k \, ,$$

where m is the largest integer smaller than $n/2p$.

Thus, the attractivity of the prototype states falls sharply if p becomes of the order of $n/2$; correlatively, the number of stable nonprototype states increases.

If the prototype states are not orthogonal, no general result can be stated but an order of magnitude of the average minimum attractivity is given by $n/2r$, where r is the rank of the prototype vector family.

E. Illustration of the projection rule properties

In the following examples we use a small number of neurons, which allows us to study exhaustively the evolutions in state space. This restriction is not a loss of generality since the efficiency of the projection rule (relation 6) is independent of the number of neurons. The threshold vector $\boldsymbol{\theta}$ is taken equal to zero.

The first two examples (Fig. 3) aim at visualizing all possible evolutions in state space; for this reason, we consider eight neurons only. The following examples (Fig. 4) illustrate the results obtained in the previous paragraph about the attractivity of the prototype states, for a network of 16 neurons.

In Fig. 3(a) five linearly dependent prototype vectors are memorized. The rank of matrix Σ is 4. As a basic property of the projection rule the prototype states are all stable and attractors. Several nonprototype stable states arise. It should be noticed that the same network (with obviously the same graph of evolution in state space) would have been obtained with only the first four independent prototypes, since the projection matrix is the same.

TABLE I. (a) Projection rule: Error correction, journal titles. Number of neurons, $n=180$; number of prototypes, $p=60$. The meaning of the two line groups on the right column is the following: the first line of each group is the initial state in which the network is set and the second line is the stable state which it reaches. (b) Projection rule: information retrieval, telephone book. Number of neurons, $n=240$; number of prototypes, $p=24$.

60 PROTOTYPE STATES	ERROR CORRECTION
PHYSICA	JOURNEL DE PHISIQUE
JOURNAL DE PHYSIQUE	JOURNAL DE PHYSIQUE
PHYSICAL REVIEW LETTERS	
INORG. CHIM. ACTA ARTIC. LETT.	INORG. CHIM. ACTA ARTISTIC
JOURNAL OF STATISTICAL PHYSICS	INORG. CHIM. ACTA ARTIC. LETT.
SCIENCE	
BULLETIN OF MATH. BIOPHYSICS	BELLETIN OF MITH. BOIPHISIYCS
BIOLOGICAL CYBERNETICS	BELLETIN OF MITH. BKOPHYSICS
. . .	

(a)

24 PROTOTYPE STATES

ARONDEL	FRANCIS	36,	RUE DES ECOLES	PARIS 5 3362550
BOUCHER	JULIEN	5,	AV. DU LAC	PARIS 4 9243189
BOUCHER	ANNIE	33,	RUE LOUPIDELOUPE	PARIS 7 0456791
ISALISE	ALFRED	68,	BD. MASSENA	PARIS 7 5002736
JEANNOT	CORINNE	13,	RUE DE LA GAITE	PARIS 6 3789720
LAROSE	JEANNOT	97,	RUE LORIENT	PARIS 5 1236785
. . .				

INFORMATION RETRIEVAL

BOUCHER	JULIEN			PARIS 4
BOUCHER	JULIEN	5,	AV. DU LAC	PARIS 4 9243189
ISALISE	FRANCIS			PARIS 7 5002736
ISALISE	ALFRED	68,	BD. MASSENA	PARIS 7 5002736
LAROSE		97,	RUE LORIENT	PARIS 5
LAROSE	JEANNOT	97,	RUE LORIENT	PARIS 5 1236785

(b)

In Fig. 3(b) three orthogonal prototype states are memorized. The prototype states are obviously stable and in this case they have the highest attractivity. The state space is regularly partitioned. The second attractors are the thresholded columns of the matrix ("symmetric states" mentioned by Amit *et al*;[6] a symmetric state $\boldsymbol{\sigma}^s$ is such that the absolute values of all the inner products $(\boldsymbol{\sigma}^s)^T\boldsymbol{\sigma}^k$ are equal or zero for all k).

Figure 4 illustrates the case in which orthogonal prototype vectors are memorized. The stable states can be gathered in a small number of classes of equal attractivity. The prototypes and their negatives always belong to the classes of highest attractivity. Histograms of the percentage of attracted states versus the Hamming distance to the attractors show that, as expected, all the states lying within a distance of $n/2p$ of a prototype are attracted (see Sec. II D 3).

F. Examples of applications

The examples presented below illustrate potential applications of neural networks designed after the projection rule. The family of prototype states is a set of p sentences; each of the m alphabetic characters is coded on six bits so that the total number of neurons is $n=6m$.

In Table I(a), an example of error correction is presented, showing that the correction is always successful unless the initial vector is too different from all the prototype states: the titles of 60 scientific journals have been chosen as prototype patterns; if the network is initially given a distorted version of a title it will generally evolve in a few iterations (three on the average) until it reaches the correct title; thus the correction process is successful except if the data is too distorted. The last lines show an example of such a situation, in which the network evolves to a nonprototype stable state.

Table I(b) shows another application in which we use the ability of the network to retrieve information from truncated data: each prototype pattern is an item in a telephone book; it contains the name of a person, his address and his phone number; if the neural network is given a part of this information such as, for instance, the name and district number, the system retrieves the complete information.

It should be mentioned that it is possible to use such a network to perform associations, by using a particular construction of the prototype vectors suggested by Kohonen[19] for linear associative memories; during the learning phase, each pattern vector $\mathbf{e}^k$ of a particular class r is presented together with the vector code $\mathbf{c}^r$ of this class. The state vectors $\boldsymbol{\sigma}$ include two distinct fields, one of which is devoted to the pattern vector to be classified $\mathbf{e}$ and the other to the code vector $\mathbf{c}$ of the class. Starting from a vector $\boldsymbol{\sigma}=\binom{\mathbf{e}}{\mathbf{c}}$ where $\mathbf{e}$ is a distorted version of a memorized vector $\mathbf{e}^k$ and $\mathbf{c}$ any vector, the network will hopefully evolve to vector $\boldsymbol{\sigma}^k=\binom{\mathbf{e}^k}{\mathbf{c}^r}$ thus performing the association of $\mathbf{e}$ to $\mathbf{c}^r$. This procedure has been proved very successful for applications such as pattern recognition,[20] since it allows an effective use of the nonprototype-stable states and greatly enhances the associative properties of the network.

III. A GENERALIZED LEARNING RULE FOR ASSOCIATION

A. Designing a neural network satisfying a given set of dynamic behavior constraints

The problem which is addressed in the present section goes beyond the mere stability of the prototype states: we show that it is possible to design neural networks which satisfy a given set of constraints; for instance, one may wish to design a neural network exhibiting a given set of stable states, a given set of transitions, and/or a given set of cycles.

1. Formulation of the problem

According to relation (1), the dynamics of the network is governed by

$$v_i(t)-\theta_i\neq 0 \Longrightarrow \sigma_i(t+\tau)=\mathrm{sgn}[v_i(t)-\theta_i]\ ,$$
$$v_i(t)-\theta_i=0 \Longrightarrow \sigma_i(t+\tau)=\sigma_i(t)\ ,$$

with $v_i(t)=\sum_{j=1}^{n} C_{ij}\sigma_j(t)$.

Suppose that we want to compute matrix C so as to impose p one-step transitions in state space,

$$\boldsymbol{\sigma}^k \rightarrow (\boldsymbol{\sigma}')^k,\quad k=1,2,\ldots,p\ . \tag{9}$$

Notice that if $(\boldsymbol{\sigma}')^k=\boldsymbol{\sigma}^k$ for all k, the problem reduces to imposing the stability of the prototype states. The problem can be expressed as a system of np inequalities,

$$\left[\sum_{j=1}^{n} C_{ij}\sigma_j^k-\theta_i\right](\sigma')_i^k>0,\quad 1\le i\le n \text{ and } 1\le k\le p \tag{10}$$

from which the elements C_{ij} of matrix C should be computed. In terms of magnetic systems, these inequalities express simply the fact that spin σ_i^k will flip into the direction of its local field,

$$\sum_{j=1}^{n} C_{ij}\sigma_j^k-\theta_j,\quad 1\le i\le n\ ,$$

to give spin $(\sigma')_i^k$. Instead of trying to solve system (10), we transform it into a linear problem,

$$\sum_{j=1}^{n} C_{ij}\sigma_j^k-\theta_i=A_i^k(\sigma')_i^k,\quad \text{for all } i \text{ and } k \tag{11}$$

where A_i^k is an arbitrary positive coefficient. This system of equations can be reduced to a matrix system,

$$C\boldsymbol{\sigma}^k-\boldsymbol{\theta}=A^k(\boldsymbol{\sigma}')^k,\quad \text{for all } k$$

where A^k is a positive diagonal matrix. It may be further reduced to a single matrix equation

$$C\Sigma=F\ , \tag{12}$$

where

$$\Sigma=(\boldsymbol{\sigma}^1,\boldsymbol{\sigma}^2,\ldots,\boldsymbol{\sigma}^p)\ ,$$
$$F=(\mathbf{f}^1,\mathbf{f}^2,\ldots,\mathbf{f}^p)\ ,$$

with $\mathbf{f}^k=A^k(\boldsymbol{\sigma}')^k+\boldsymbol{\theta}$.

2. *Solution of Eq. (12): the associating rule*

Equation (12) does not always have an exact solution.

(i) If $F\Sigma^I\Sigma = F$, Eq. (12) has an exact solution, the general form of which is

$$C = F\Sigma^I + B(I - \Sigma\Sigma^I) , \tag{13}$$

where Σ^I is the Moore-Penrose pseudoinverse, and B is an arbitrary (n,n) matrix. In the present paper we take $B=0$, so that we have the following learning rule:

$$C = F\Sigma^I . \tag{14}$$

However, in some cases, B provides a degree of freedom which has already proved fruitful for the modeling of biological mechanisms.[13] Since $\Sigma^I\Sigma$ is the orthogonal projection matrix into the subspace spanned by the rows of Σ, the above condition $F\Sigma^I\Sigma = F$, which can be written equivalently $\Sigma^I\Sigma F^T = F^T$, means that the rows of F are linear combinations of the rows of Σ.

In the particular case where the vectors σ^k are linearly independent, as mentioned in Sec. II, the pseudoinverse takes the form

$$\Sigma^I = (\Sigma^T\Sigma)^{-1}\Sigma^T \text{ and } \Sigma^I\Sigma = I .$$

(ii) If $F\Sigma^I\Sigma \neq F$ there is no exact solution but $C = F\Sigma^I$ is the matrix which minimizes the Euclidean norm of the error matrix $C\Sigma - F$.

In the first case, when an exact solution exists, there is still an infinity of possible matrices C, satisfying the required set of constraints (9), depending on F. The computation of the coupling matrix C may be further simplified by the following argument: for a given θ, it is possible to find λ such that

$$-\lambda < \theta_i < \lambda, \text{ for all } i . \tag{15}$$

It can be easily shown that a set of positive diagonal matrices A^k exists if we impose

$$F = \lambda\Sigma', \ \Sigma' = [\sigma'^1, \sigma'^2, \ldots, \sigma'^p] ,$$

so that matrix C reduces to

$$C = \lambda\Sigma'\Sigma^I . \tag{16}$$

This rule will be referred to in the following as the *associating learning rule,* because it allows us to impose that the network perform the associations $\sigma^k \rightarrow (\sigma')^k$ for

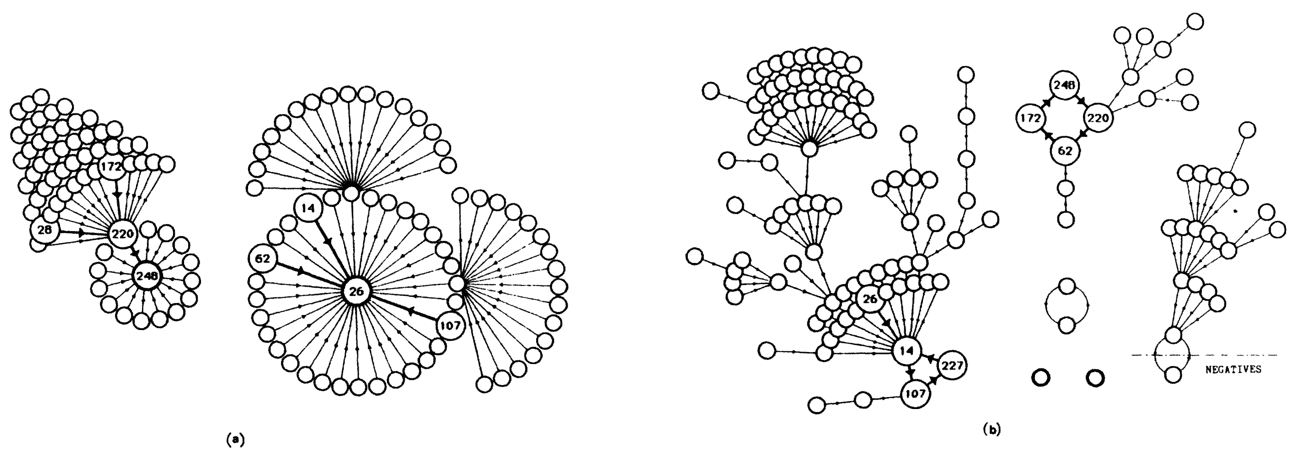


FIG. 5. Associating rule: Evolution in state space. These two examples show the evolution in state space of a network of eight neurons. Heavy lines represent the imposed one step transitions. (a) Classification. Attractors (248) and (26) code classes. The states (172), (28), etc. are given as examples of elements of these classes. Matrix Σ,

(28) (172) (220) (248) (14) (62) (107) (26) .

Matrix Σ',

(220) (220) (248) (248) (26) (26) (26) (26) .

(b) Cycles. Two cycles,

(248)–(220)–(62)–(172)–(248)

(14)–(107)–(227)–(14)

and the transition (26)—(14) are imposed during the learning phase. Matrix Σ,

(248) (220) (62) (172) (26) (14) (107) (227) .

Matrix Σ',

(220) (62) (172) (248) (14) (107) (227) (14) .

It should be noticed that nonimposed cycles of length two appear.

TABLE II. Associating rule: Classification. Books classified by editor's names. Number of neurons, $n=240$; number of imposed constraints, $p=66$.

60 TITLES	6 EDITORS
CONTENT ADDRESSABLE MEMORIES	SPRINGER SERIES IN INFORMATION SCIENCES
PROBABILITY AND STATISTICAL PHYSICS	NORTH HOLLAND SERIES IN APPLIED MATH
MATHEMATICAL METHODS IN TECHNOLOGY	DUNOD INFORMATIQUE PHASE SPECIALISEE
BASES DE DONNEES METHODES PRATIQUES	WILEY SERIES IN MATHEMATICAL STATISTICS
TELEINFORMATIQUE	ADDISON WESLEY MICROBOOKS SERIES
TECHNIQUES STATISTIQUES	DUNOD TECHNIQUE MATHEMATIQUES
APPLIQUEES	
...	

CLASSIFICATION
PROFABILIGY AND STATDSTICAL PHYSICS
SPRINGER SERIES IN INFORMATION SCIENCES
TECHNICAL STATISTICS
DUNOD INFORMATIQUE PHASE SPECIALISEE
BASIS OF DATAS METHODS PRACTICAL
DUNOD YECHKHQUEEPATPELA AQPEB APHTIDU ES

$k=1,2,\ldots,p$. If $\Sigma'=\Sigma$, disregarding the scaling factor λ, relation (16) reduces to the projection rule.

3. *Examples of networks designed after the associating learning rule*

Here we present several examples obtained when the network is designed after the associating rule $C=\Sigma'\Sigma^I$ with $\theta=0$ which leaves the largest margin between $-\lambda$ and $+\lambda$. By imposing the evolution in one step of a given set of states, this rule allows us to impose specific evolutions in state space such as a transient sequence of states leading to an attractor or a cycle.

Transient sequences of states leading to an attractor. These kinds of associations may be useful in two cases.

(i) Classification. In Fig. 5(a) we present an example in which several steps have been imposed; after the learning period the network will associate state (172) to state (220) and state (220) to the stable attractor state (248), etc. Supposing that states (14), (28), (62), (107), (172), and (220) are prototype patterns and that states (26) and (248) code the classes, this allows us to make pattern recognition or classification.

(ii) Suppression of undesired non-prototype attractor states. A network designed after either the projection or the associating rule may exhibit undesired attractors which can be eliminated by imposing extra transitions from these undesired attractors to proper attractors (see, for instance, an example of application in Sec. III B).

Cycles. In Fig. 5(b) we present an example in which two cycles have been imposed. Interestingly, simulations indicate that if cycles of at most m steps are imposed, no secondary cycles of length larger than m appear.

B. Application: use of the network for classification

In this section we want the neural network to behave as an associative memory which, after learning several examples of each class of information, is able to classify faithfully an incomplete or distorted information. As mentioned in Sec. III A, this is possible with the associating rule. An example of classification of books by editor name is given in Table II. During the learning phase each book title was associated with the name of its editor which is itself made stable. After learning, the network is able to retrieve the name of the editor, even if it is given a distorted version of the title of a book. One can notice that the last example leads to a nonimposed attractor which may be considered undesirable. By adding the following imposed transition,

$$\text{DUNOD YECHKHQUEEPATPELA AQPEB APHTIDU ES} \rightarrow \text{DUNOD INFORMATIQUE PHASE SPECIALISEE},$$

we eliminate it successfully while the other results remain unchanged.

CONCLUSION

In this paper, we have presented a general formulation of the design of neural networks. Starting from an analysis of the behavior of the network, we have derived an associating rule which allows us to impose a given set of dynamic properties: stable states and/or valleys and/or cycles. The associating rule has been used for classification purposes. In the particular case where only stable states are imposed, the associating rule reduces to the projection rule, which allows us to memorize and retrieve faithfully a set of prototype patterns. Several analytical results related to the decision mechanism, the nonprototype attractors, the absence of cycles, and the attractivity of the prototype states have been derived. These new design rules provide flexible tools for performing high-level information-processing functions, such as memorization and association.

ACKNOWLEDGMENT

We wish to thank J. L. Zorer for numerical simulations performed on the CIRCE computers (Orsay).

APPENDIX A

We consider the free evolution of a network, designed after the projection rule, from a state $\sigma=\sigma(t)$ to a state $\sigma'=\sigma(t+\tau)$ in one parallel iteration: first, the network computes the potential vector $\mathbf{v}$ by the relation

$$\mathbf{v}=C\sigma ;$$

then, decision rule (1), with $\theta=0$, sets the network into the state σ' such that

$$\sigma_i'=\mathrm{sgn}(v_i), \quad \text{if } v_i\neq 0$$

$$\sigma_i'=\sigma_i, \quad \text{if } v_i=0 .$$

One has

$$(\sigma')^T\mathbf{v}=\sum_i |v_i| .$$

Consider a vector $\sigma^*\neq\sigma'$, the difference between them arising from at least one component j (with $v_j\neq 0$). One has

$$(\sigma^*)^T\mathbf{v}=\sum_i |v_i|\sigma_i^*\sigma_i' .$$

Since at least the jth term of the sum is equal to $-|v_j|$, one has

$$\sum_i |v_i|\sigma_i^*\sigma_i' < \sum_i |v_i| ,$$

so that $(\sigma^*)^T\mathbf{v}<(\sigma')^T\mathbf{v}$. Since σ and σ' have the same norm ($\sqrt{n}$), the latter relation is equivalent to

$$||\sigma^*-\sigma||>||\sigma'-\mathbf{v}|| . \tag{A1}$$

Therefore

(i) let us consider a state $\sigma(t)$; if all the components of $\mathbf{v}=C\sigma(t)$ are nonzero, the next state $\sigma(t+\tau)$ will be the vector of $\{-1,+1\}^n$ which is closest to $\mathbf{v}$ (with respect to the Euclidean distance).

(ii) If q of the components of $\mathbf{v}$ are zero, $\sigma(t+\tau)$ will be one of the 2^q vectors of $\{-1,+1\}^n$ which are closest to $\mathbf{v}$, the components of $\sigma(t+\tau)$ corresponding to components of $\mathbf{v}$ equal to zero being the same as those of $\sigma(t)$.

(iii) Consequently, if σ is among the vectors $\{-1,+1\}^n$ which are closest to $C\sigma$, then σ is stable.

APPENDIX B

We consider the free evolution of a network, designed after the projection rule, from a state $\sigma=\sigma(t)$ to a state $\sigma'=\sigma(t+\tau)$ in one parallel iteration. We shall show that the energy defined by

$$E(\sigma)=-(\tfrac{1}{2})\sigma^T C\sigma$$

is an ever-decreasing function, that is

$$E(\sigma')<E(\sigma), \quad \text{if } \sigma'\neq\sigma .$$

Since $C\sigma'$ is the orthogonal projection of σ' on the subspace L_p spanned by the prototype vectors, it is the vector belonging to L_p which is closest to σ'. Therefore $||\sigma'-C\sigma'||<||\sigma'-C\sigma||$ if $\sigma'\neq\sigma$. Moreover, relation (A1) gives

$$||\sigma'-C\sigma||<||\sigma-C\sigma|| ,$$

thus

$$||\sigma'-C\sigma'||<||\sigma-C\sigma|| .$$

Since σ and σ' have the same norm ($\sqrt{n}$), the latter relation is equivalent to $(\sigma')^T C\sigma'>\sigma^T C\sigma$; therefore $E(\sigma')<E(\sigma)$.

APPENDIX C

Let $\{\sigma^k\}$ be a family of p orthogonal prototype vectors. Matrix C is calculated after the projection rule which, in this case, reduces to Hebb's rule (3). Let σ be a state vector different from the prototype ones. We investigate the evolution of the system when started in state σ. We apply C to σ,

$$C\sigma=(1/n)\sum_{k=1}^{p}\sigma^k(\sigma^k)^T\sigma .$$

Thus $C\sigma$ is a linear combination of the prototype states, the coefficients of which are the inner products of σ^k and σ. Since the components of these vectors are -1's and $+1$'s, one has

$$(\sigma^k)^T\sigma=n-2H(\sigma^k,\sigma) ,$$

where $H(\sigma^k,\sigma)$ is the Hamming distance between states σ^k and σ. In the following we denote $H(\sigma^k,\sigma)$ by H_k. Therefore one has

$$C\sigma=(1/n)\sum_{k=1}^{p}(n-2H_k)\sigma^k .$$

To investigate the attractivity of the prototype state σ^m, we try to find a sufficient condition for the system to evolve from state σ to state σ^m in one iteration.

It results from the evolution rule (1) of the network that, if each component of $C\sigma$ has the same sign as the corresponding component of σ^m, the network evolves from state σ to state σ^m in one iteration,

$$(C\sigma)_i\sigma_i^m=(1/n)(n-2H_m) + (1/n)\sum_{\substack{k=1\\k\neq m}}^{p}(n-2H_k)\sigma_i^k\sigma_i^m>0, \quad \text{for all } i .$$

This condition is satisfied if

$$\left| \sum_{\substack{k=1 \\ k\neq m}}^{p} (n-2H_k)\sigma_i^k\sigma_i^m \right| < n-2H_m, \quad \text{for all } i\ . \tag{C1}$$

An upper limit of the left-hand term can be found,

$$\left| \sum_{\substack{k=1 \\ k\neq m}}^{p} (n-2H_k)\sigma_i^k\sigma_i^m \right| \leq \sum_{\substack{k=1 \\ k\neq m}}^{p} |(n-2H_k)|\ .$$

After the triangular inequality, σ^k and σ^m being two orthogonal vectors, one has

$$H_k \leq H_m + n/2\ ,$$

$$n/2 \leq H_k + H_m\ ,$$

which imply

$$|n-2H_k| \leq 2H_m\ . \tag{C2}$$

Therefore

$$\left| \sum_{\substack{k,m=1 \\ k\neq m}}^{p} (n-2H_k)\sigma_i^k\sigma_i^m \right| \leq 2H_m(p-1)\ .$$

Consequently, if the condition $2H_m(p-1) < n-2H_m$ is satisfied, relation (C1) will be verified, thus the network will certainly evolve from state σ to state σ^m. The last relation implies that

$$H_m < n/2p\ . \tag{C3}$$

It can be checked from relation (C2) that if a state lies within a distance of $n/2p$ from a given prototype state, its distance from any other prototype state is larger than $n/2p$,

$$n-2H_k < 2H_m < n/p, \quad \forall k\neq m$$

hence

$$H_k > n/2p\ .$$

In summary, we have shown that, if a state σ lies within a distance of $n/2p$ of a prototype state σ^m: (i) σ^m is the nearest prototype state, (ii) the network will evolve from state σ to state σ^m in one iteration. It should be noticed that $n/2p$ is a lower limit of the size of basin of attraction of a prototype state; starting states lying at larger distances may lead the network to that prototype state. it can be shown similarly that any state lying within a distance of $n/2p$ of a state $-\sigma^m$ will converge to that state in one iteration.

[1]F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms* (Spartan, Washington, D.C., 1962).

[2]M. Minsky and S. Papert, *Perceptrons* (MIT Press, Cambridge, 1969).

[3]W. A. Little, Math. Biosci. **19**, 101 (1974).

[4]J. J. Hopfield, Proc. Natl. Acad. Sci. **79**, 2554 (1982).

[5]P. Peretto, Biol. Cybern. **50**, 51 (1984).

[6]D. J. Amit, H. Gutfreund, and H. Sompolinsky, Phys. Rev. A **32**, 1007 (1985).

[7]F. V. Feigelman and L. B. Ioffe, Europhys. Lett. **1**, 197 (1986).

[8]W. Kinzel, Z. Phys. B **60**, 205 (1985).

[9]L. Personnaz, I. Guyon, and G. Dreyfus, J. Phys. Lett. **46**, L359 (1985); in *Disordered Systems and Biological Organization*, edited by E. Bienenstock *et al.* (Springer, New York, 1986).

[10]I. Kanter and H. Sompolinsky, in *Proceedings of the Heidelberg Colloquium on Glassy Dynamics and Optimization, Heidelberg, 1986*, edited by J. L. van Hemmen and I. Morgenstern (Springer, New York, in press).

[11]N. Parga and M. A. Virasoro (unpublished).

[12]G. Toulouse, S. Dehaene, and J. P. Changeux, Proc. Natl. Acad. Sci. **83**, 1695 (1986).

[13]L. Personnaz, I. Guyon, G. Dreyfus, and G. Toulouse, J. Stat. Phys. **43**, 411 (1986).

[14]D. Hebb, *The Organisation of Behavior* (Wiley, New York, 1949).

[15]L. N. Cooper, F. Liberman, and E. Oja, Biol. Cybern. **33**, 9 (1979).

[16]K. G. Beauchamp, *Applications of Walsh and Related Functions* (Academic, New York, 1984).

[17]A. Albert, *Regression and the Moore-Penrose Pseudoinverse* (Academic, New York, 1972).

[18]F. Fogelman, E. Goles, and G. Weisbuch, Discrete Appl. Math. **6**, 95 (1983).

[19]T. Kohonen, *Self Organization and Associative Memories* (Springer, Berlin, 1984).

[20]L. Personnaz, I. Guyon, P. Siarry, and G. Dreyfus, *Proceedings of the Heidelberg Colloquium on Glassy Dynamics and Optimization, Heidelberg, 1986*, edited by J. L. van Hemmen and I. Morgenstern (Springer, New York, in press).