Recognition rates of the Hebb rule for learning Boolean functions

F. Vallet and J-G. Cailton
Laboratoire Central de Recherches, Thomson-CSF, 91404 Orsay CEDEX, France
(Received 13 March 1989; revised manuscript received 31 August 1989)

We study the Hebb rule for learning several Boolean functions (random and linearly separable functions) defined on the hypercube of dimension N. Learning and generalization rates are derived in the $N\to\infty$ limit versus $\alpha=P/N$, where P is the number of learned patterns. In the linearly separable case, the generalization rate grows monotonically from $\frac{1}{2}$ to 1, whereas the learning rate first decreases from 1 to a minimum value, and then increases again towards 1. This result is interpreted as an interference phenomenon, like in the learning for associative memories implemented with the same rule. Comparisons are then made with the case of random Boolean functions, associative memories, and their clipped version. The behavior of the Hebb rule is decomposed in two distinct contributions, referred to as the "rote" and the "conceptual" learnings. Illustrative numerical simulations are given.

I. INTRODUCTION

In the area of automatic learning (memorization of patterns, learning of rules, etc.), neural networks have already provided quite interesting performances.[1] As a very simple neuronal model, the perceptron architecture has been widely investigated by neurobiologists, physicists, and mathematicians. A perceptron is a neuronal architecture that is defined by only one layer of adaptive synaptic weights and that performs a simple mathematical transformation on an input vector. As we are interested here in learning Boolean functions, only one output bit is required in the architecture. According to the perceptron-type model,[2-4] the output bit $\mathcal{P}_W$ of a perceptron is given versus the input vector $|X\rangle$ by

$$|X\rangle\to\mathcal{P}_W(|X\rangle)=\mathrm{sgn}(\langle W|X\rangle)=\pm 1\ , \tag{1}$$

where $|W\rangle$ is the perceptron weight vector ($\langle W|$ is its transposed vector, so that $\langle W|X\rangle$ is the inner product).

In this paper we address the issue of the performances of the Hebb rule, implemented on a perceptron-type network, for the learning of several Boolean functions defined on the N-hypercube $\{-1,+1\}^N$.

The general problem addressed here is to approximate, with a perceptron of the form (1), a given Boolean function $\mathcal{B}$, from the knowledge of a set L of patterns $L=\{|X_\mu\rangle\}_{\mu=1,\ldots,P}$ (called the learning set, and composed of $P=\alpha N$ patterns) with their corresponding desired output values $\mathcal{B}(|X_\mu\rangle)=\pm 1$.

We define two recognition rates. The learning rate is defined as the probability for a learned pattern $|X_\mu\rangle$ to be recognized well [i.e., to verify $\mathcal{P}_W(|X_\mu\rangle)=\mathcal{B}(|X_\mu\rangle)$]; and the generalization rate as the probability for an arbitrary pattern $|X\rangle$ taken randomly in $\{-1,+1\}^N$ to be recognized well.

The Hebb rule gives the simplest way for correlating the effective output $\mathrm{sgn}\langle W|X_\mu\rangle$ of the perceptron to the desired one $\mathcal{B}|X_\mu\rangle$; it is defined by its weight vector $|W_{\mathrm{Hebb}}\rangle$, computed from the learning set L:[5]

$$|W_{\mathrm{Hebb}}\rangle=\sum_{\mu=1}^{\alpha N}|X_\mu\rangle\mathcal{B}|X_\mu\rangle\ . \tag{2}$$

In this paper, we will study the recognition performances of this Hebb solution in two cases:

(1) $\mathcal{B}$ is a linear separable Boolean function ($\mathcal{B}|X\rangle=\mathrm{sgn}\langle B|X\rangle$, where $|B\rangle\in\mathbb{R}^N$ is a given weight vector defining the Boolean function), and

(2) $\mathcal{B}$ is a random function ($\mathcal{B}|X\rangle=\pm 1$, the two values being chosen randomly and independently, with equal probability $\frac{1}{2}$).

We do not address here the general problem of the optimal capacity of perceptrons for learning Boolean functions,[6,7] or the performances of more complicated learning rules such as the pseudo-inverse[8-10] or iterative rules.[11,3] For example, results about the relationship between the learning and the generalization rates have been given in the literature[12] which provide a lower bound for the generalization rate when the learning rate is good enough. These results are of course in agreement with those presented here, for which the learning rate is never optimal (for a given α), because of the raw form of the Hebb rule.

The input patterns belong to the hypercube of dimension N. The learning set is composed of $P=\alpha N$ random patterns $\{|X_\mu\rangle\}_{\mu=1,\ldots,P}$ whose components are randomly and independently taken in $\{+1,-1\}$, with equal probability $\frac{1}{2}$. For the generalization, the test patterns are chosen with the same rule.

II. THE TECHNIQUE OF DERIVATION OF THE RECOGNITION RATES

The rates we are looking for in this paper are averaged rates. That is to say that for a given Boolean function $\mathcal{B}$, we want to evaluate the probability over all learning sets and all test patterns, that this pattern is recognized well. We will decompose this problem into two distinct steps: we derive first the probability, over all learning sets, for a

given pattern $|X\rangle$ to be recognized well. We then average this recognition rate over all the $|X\rangle$'s.

If $|X\rangle$ is a test vector, the Hebb solution gives, for the output bit,

$$\mathrm{sgn}(\langle X|W_{\mathrm{Hebb}}\rangle)=\mathrm{sgn}\left[\sum_{\mu=1}^{\alpha N}\langle X|X_\mu\rangle\mathcal{B}|X_\mu\rangle\right] \tag{3}$$

so that this pattern is recognized well by the Hebb solution if this expression has the same sign as $\mathcal{B}|X\rangle$, that is to say if

$$\sum_{\mu=1}^{\alpha N} z^{\mu}_{|X\rangle}\geq 0 \quad \text{where } z^{\mu}_{|X\rangle}=\langle X|X_\mu\rangle\mathcal{B}|X_\mu\rangle\mathcal{B}|X\rangle\ . \tag{4}$$

We then deduce the mean recognition rate $R_{|X\rangle}$ for this particular vector $|X\rangle$, by deriving the probability averaged over all the Hebb vectors $|W_{\mathrm{Hebb}}\rangle$ (each one defined by a particular choice of the learning set L given by the P randomly chosen $|X_\mu\rangle$'s) that the Hebb solution gives the right answer:

$$R_{|X\rangle}(\alpha)=\mathrm{Prob}_{\{|X\rangle_\mu\}_{\mu=1,\ldots,P}}\left[\sum_{\mu=1}^{\alpha N} z^{\mu}_{|X\rangle}\geq 0\right], \tag{5}$$

and then the average recognition rate is derived by integrating this rate over all the possible values of $|X\rangle$:

$$R(\alpha)=\frac{1}{2^N}\sum_{|X\rangle}R_{|X\rangle}(\alpha)\ . \tag{6}$$

If the learning set L is chosen independently of $|X\rangle$, we obtain for $R(\alpha)$ the mean generalization rate noted hereafter $G(\alpha)$; if L always contains $|X\rangle$ [say $|X_1\rangle=|X\rangle$ in (3)–(5)] we obtain the mean learning rate [in that case, we have $z^{1}_{|X\rangle}=N$ in (5)].

III. THE HEBB RULE FOR LEARNING LINEARLY SEPARABLE FUNCTIONS

We are interested here by the linearly separable case: $\mathcal{B}|X\rangle=\mathrm{sgn}\langle B|X\rangle$, and focus our attention on two limit cases, for which the recognition rates can be derived exactly, and only depend, in the $N\to\infty$ limit, on the ratio $\alpha=P/N$ (P is the number of patterns taken into account during the learning phase).

(1) The first one corresponds to the situation in which only one input bit is taken into account for the evaluation of the Boolean function, that is to say that $|B\rangle=(1,0,0,\ldots,0)$; this case will be referred to as the "easy case" (we will see later that the recognition rates are the highest for this problem). The average learning and generalization rates are found to be, respectively (see Sec. 1 of the Appendix),

$$L_{\mathrm{easy}}(\alpha)=\tfrac{1}{2}\mathrm{erfc}\left[\frac{-1}{\sqrt{2\alpha}}-\sqrt{\alpha/2}\right], \tag{7}$$

$$G_{\mathrm{easy}}(\alpha)=\tfrac{1}{2}\mathrm{erfc}(-\sqrt{\alpha/2})\ , \tag{8}$$

where we use the complementary error function defined as follows:

$$\mathrm{erfc}(x)=\frac{2}{\sqrt{\pi}}\int_x^{\infty}e^{-u^2}du\ . \tag{9}$$

(2) In the second one, all the input components of $|X\rangle$ have an equivalent role, and $|B\rangle$ is of the following form:

$$|B\rangle=(B_1,\ldots,B_1;B_2,\ldots,B_2;\ldots;B_q,\ldots,B_q)\ , \tag{10}$$

each value $B_i\in\mathbb{R}$ being repeated N/q times, so that $|B\rangle\in\mathbb{R}^N$ (q is a fixed integer value, whereas N will tend towards infinity). Here, the result is independent of the specific form of $|B\rangle$ (it does not depend on q, nor on the B_i's). This case will be referred to as the "hard case." The average learning and generalization rates are found to be, respectively (see Sec. 2 of the Appendix or Ref. 5),

$$L_{\mathrm{hard}}(\alpha)=\frac{1}{\sqrt{\pi}}\int_0^{\infty}e^{-u^2}\mathrm{erfc}\left[-u\sqrt{2\alpha/\pi}-1/\sqrt{2\alpha}\right]du\ , \tag{11}$$

$$G_{\mathrm{hard}}(\alpha)=\frac{1}{\sqrt{\pi}}\int_0^{\infty}e^{-u^2}\mathrm{erfc}(-u\sqrt{2\alpha/\pi})du\ . \tag{12}$$

We give in Sec. 5 of the Appendix the asymptotic limits of these rates for $\alpha\to\infty$ and $\alpha\to 0$. We stress that in both cases

$$L(0)=L(\infty)=G(\infty)=1,\quad G(0)=\tfrac{1}{2}\ . \tag{13}$$

These rates have been plotted in Fig. 1. We remark that the generalization rates grow monotonically from $\frac{1}{2}$ to 1, whereas the learning rates first decrease from 1 to a minimum value and then tend again towards 1 when α goes toward infinity. The first result shows that the direction of the Hebb vector $|W_{\mathrm{Hebb}}\rangle$ tends towards the good solution $|B\rangle$ when α goes to infinity. Concerning the learning rate, for small values of α, although the Hebb vector $|W_{\mathrm{Hebb}}\rangle$ is still far from the real value $|B\rangle$, it is well suited for the special vectors belonging to the learning set, whereas for intermediate values of α, it is still far from this good solution (the learning patterns are not numerous enough to evaluate $|B\rangle$ correctly because of incomplete information), but gives rise to interferences between the learned patterns (as for the associative memories implemented with the same rule) and causes confusion. The terms proportional to $\sqrt{\alpha}$ in Eqs. (7), (8), (11), and (12) correspond to the generalization effect by which the system learns the coherence between all the examples it learns ($|W_{\mathrm{Hebb}}\rangle\to|B\rangle$ when α grows), and the term proportional to $1/\sqrt{\alpha}$ in (7) and (11) (present only in the learning rates) corresponds to the "rote" learning, which provides a high contribution when α is small, and tends to disappear when $\alpha\to\infty$ because of the interference effects between the learned patterns. The generalization term $\sqrt{\alpha}$ reflects the fact that the system gets a correct view of the problem (a "conceptual" one) which improves when more examples are presented, whereas the $1/\sqrt{\alpha}$ term corresponds to the "rote" learning, which saturates (confusion) when too many patterns are learned.

This interpretation can be illustrated in another way, by "clipping" the solution of the hard case, when the vector to be found has all its components equal to the same value $+1$: $|B_0\rangle=(1,1,\ldots,1)$. The "clipping" transformation keeps only the signs of the components of the Hebb vector:

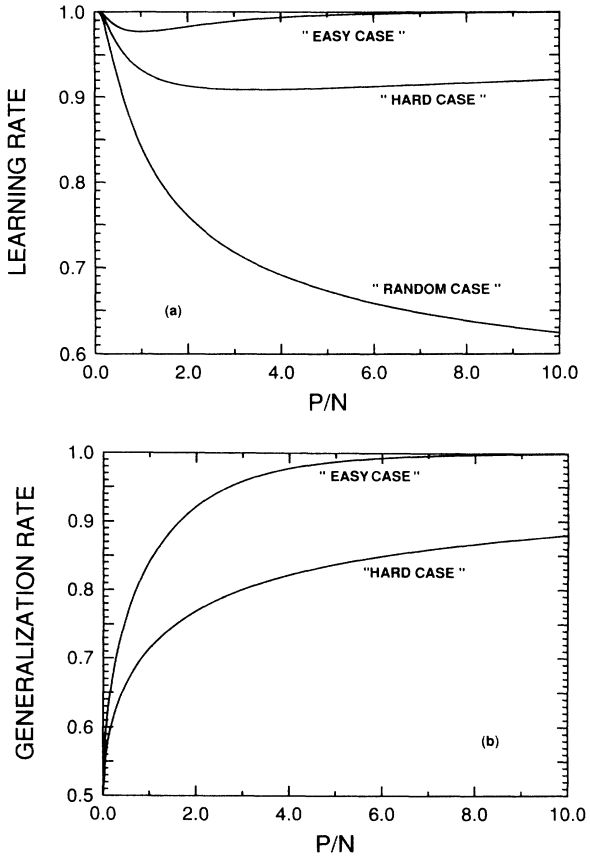

FIG. 1. Learning rate (a) and generalization rate (b) vs $\alpha=P/N$, with the Hebb rule, of two linearly separable Boolean functions. The directions to be found are $|B\rangle=(1,0,0,..,0)$ for the "easy case," and $|B\rangle=(B_1,\ldots,B_1;B_2,\ldots,B_2;\ldots;B_q,\ldots,B_q)$ for the "hard case" (each component B_i being repeated N/q times, so that $|B\rangle\in\mathbb{R}^N$; the curves do not depend on the particular values of the B_i's). The "random case" refers to the case when the output bit is no longer correlated to the input vector, and is randomly chosen.

$$|W_{\text{Hebb}}\rangle\rightarrow|W_{\text{clip}}\rangle=\text{sgn}|W_{\text{Hebb}}\rangle\ . \tag{14}$$

As $|B_0\rangle$ is invariant by this transformation, it could be expected that applying the clipping transformation on the Hebb solution improves the recognition rates. Numerical simulations (Fig. 2) show that this is not true for the learning rate in the region of small values of α. In that region, the transformation lowers the learning rate: the Hebb solution is then overfitted to the particular shape of the learning set, and does not contain enough information about the real direction $|B_0\rangle$ to be found, so that clipping only leads the solution to forget a little about the learned patterns without bringing it closer to the exact solution because of lack of information. In the region of large values of α, clipping becomes efficient because the Hebb solution is near the exact one.

Concerning $|B_{\text{easy}}\rangle=(1,0,0,\ldots,0)$, the clipping is dramatically bad, because in its new clipped form, $|B_{\text{easy}}\rangle$

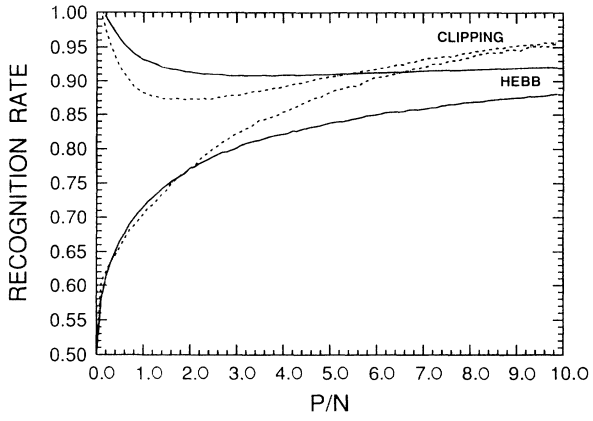

FIG. 2. Numerical simulations for the learning (upper curve) and generalization (lower curve) rates vs α, for the Hebb rule (continuous line) and its "clipped" form (dashed line) when the direction to be found is given by $|B_0\rangle=(1,1,\ldots,1)$. Two regions appear: for small values of α, the "clipping" lowers the learning performances, whereas for large values it improves it. The simulation has been done in dimension $N=400$, averaged over 100 draws. Concerning the nonclipped version, the perfect fitting with the theoretical curve of Fig. 1 is worth noticing.

becomes $(1,\pm1,\pm1,\ldots,\pm1)$ which is completely different from the direction to evaluate $|B_{\text{easy}}\rangle$. We give in Sec. 3 of the Appendix the calculation which leads to the following rates in this case (see, for example, Ref. 13):

$$L_{\text{clip}}(\alpha)=\tfrac{1}{2}\text{erfc}\left[-\frac{1}{\sqrt{\pi\alpha}}\right];\quad G_{\text{clip}}(\alpha)=\tfrac{1}{2}\ . \tag{15}$$

It is clear here that only the rote contribution can subsist [the $1/\sqrt{\alpha}$ contribution in (15)], for the clipping prevents any correct generalization from occurring.

In fact the results (11) and (12) concerning the hard case should still be exact when $|B\rangle$ has a more general form, corresponding to components which are randomly and independently chosen in $\mathbb{R}$ with a given probability distribution $p(x)$: the probability for the ith component of $|B\rangle$ to lie in $[x,x+\epsilon]$ is given by $\epsilon p(x)$. Indeed we can approximate the probability law $p(x)$ by a discrete law,

$$p(x)\sim\frac{1}{q}\sum_{i=1}^{q}\delta(x-C_i) \tag{16}$$

(δ is the Dirac function) which corresponds to the case studied here. We can then take the $q\rightarrow\infty$ limit to obtain the equality in (16). It is worth emphasizing that the easy case is not of course induced by any probability law $p(x)$.

Finally, we claim that the two cases under focus here ("easy" and "hard" cases) represent the extreme possibilities for the vector $|B\rangle$ to be learned; for example, the vector $(1,1,1,0,0,0,\ldots,0)$ gives recognition rates which lie between these two extremes (see Fig. 3).

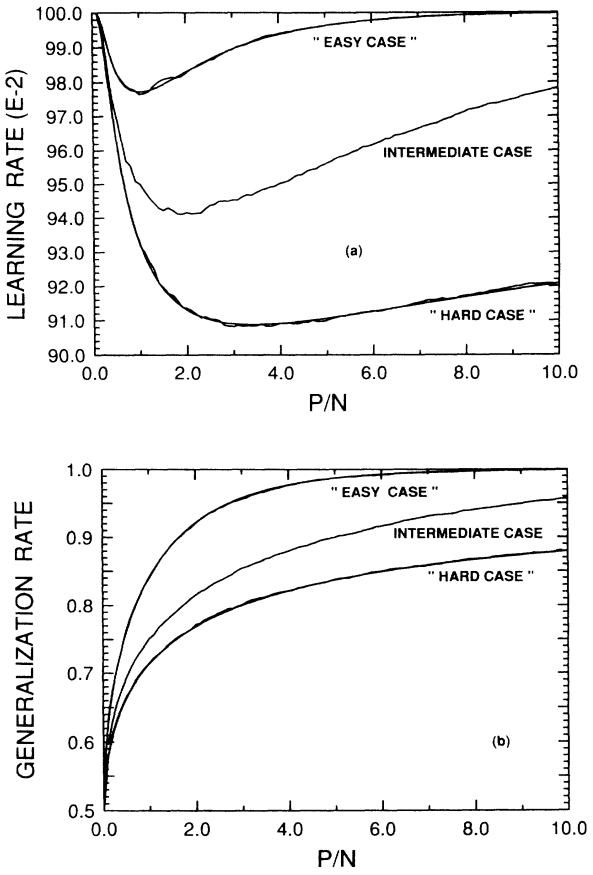

FIG. 3. Numerical simulations for the learning rate (a) and generalization rate (b) vs $\alpha=P/N$, with the Hebb rule, of three linearly separable Boolean functions, defined, respectively, by their weight vector: "easy case:" $|B_{\text{easy}}\rangle=(1,0,0,0,\ldots,0,0)$; "hard case:" $|B_{\text{hard}}\rangle=(0,1,1,5)$, each of the four components being repeated 100 times (then $N=400$, $q=4$, $B_1=0$, $B_2=B_3=1$, $B_4=5$); and "intermediate case:" $|B\rangle=(1,1,1,0,\ldots,0)$. Theoretical results have been plotted simultaneously for comparison for the two first cases (we do not have any theoretical result for the third one). Simulations are done with dimension $N=400$, and results averaged over 100 draws.

IV. THE HEBB RULE FOR LEARNING RANDOM BOOLEAN FUNCTIONS

We will see now that the "conceptual" contribution $\sqrt{\alpha/2}$ can be eliminated in the easy case (7) by choosing a random Boolean function for the output bit, so that no generalization can occur. $\mathcal{B}|X\rangle$ is no longer correlated to $|X\rangle$ and is chosen randomly in $\{-1,+1\}$, with equal probability $\frac{1}{2}$, independent of the other patterns. The derivation of the learning rate is straightforward (Sec. 4 of the Appendix) and leads to the following expression (see Fig. 1):

$$L_{\text{rand}}(\alpha)=\tfrac{1}{2}\operatorname{erfc}\left[\frac{-1}{\sqrt{2\alpha}}\right] . \tag{17}$$

No generalization rate can now be defined because of the random choice of the output values. We see clearly now the effect of interferences between learned patterns with one another, as only the rote part remains, which vanishes when α grows.

We see that the randomness of the choice for the Boolean function is well "understood" by the system. This simple remark could provide an efficient way to decide if a Boolean function given by its output bits on a given number of input patterns is linearly separable or not: draw the curve $\lambda(\alpha)$ of the learning rate versus α with the Hebb rule (each value of α must be represented by averaging the results over many different draws). If $\lambda\simeq L_{\text{rand}}$ then probably the function is not linearly separable; if $L_{\text{hard}}\leq\lambda\leq L_{\text{easy}}$ then it is certainly linearly separable.

Note that in the $\alpha\to 0$ limit, the three rates L_{hard}, L_{easy}, and L_{rand} are equivalent to $1-C\sqrt{\alpha}e^{-1/(2\alpha)}$, each learning rate having its own constant value ($C_{\text{easy}}\sim 0.1467$, $C_{\text{hard}}\sim 0.2330$, $C_{\text{rand}}\sim 0.3989$). That is to say that the learning performances of the Hebb rule for small values of α depend weakly (not by the form of the principal term, and only by the multiplicative constant of that term) on the particular shape of the learned Boolean function. But in this region of α, the coherence between the output values of B for the learning set is already detected, and the rote learning which occurs is better for the easy case than for the hard case, and better for this last one than for the "random case." This remark completes the one about the conceptual and the rote contributions in the recognition rates.

V. RELATION WITH ASSOCIATIVE MEMORIES

Concerning the easy case, we can remark that the problem is in fact strongly related to that of associative memories.[14,13,15] Indeed in that case, we look for an $N\times N$ weight matrix, denoted J, for which the learned patterns are invariant for the relaxation dynamics defined by

$$|X(t)\rangle\to|X(t+1)\rangle=\operatorname{sgn}(J|X\rangle) . \tag{18}$$

That is to say that the learned patterns $|X_\mu\rangle$ must be fixed points for the dynamics

$$\operatorname{sgn}(J|X_\mu\rangle)=|X_\mu\rangle . \tag{19}$$

The first component of Eq. (19) reads

$$\operatorname{sgn}\langle J_1|X_\mu\rangle=X_\mu^1\equiv\operatorname{sgn}\langle B_{\text{easy}}|X_\mu\rangle \tag{20}$$

(where $|J_1\rangle$ is the vector composed by the first row of the J matrix, the $N-1$ other equations corresponding to the other components are all similar and independent). We then see clearly that the learning for associative memories is equivalent to the "easy" problem. So, as when α goes to infinity, the Hebb solution for the learning of $|B_{\text{easy}}\rangle$ tends towards $|B_{\text{easy}}\rangle$ (modulo one multiplicative term), we deduce easily that the J matrix given by this rule tends towards the identity matrix (modulo a multiplicative term), and all patterns become invariant under the dynamics (18).

If we now look for a J matrix with diagonal elements which are constrained to be zero ($J_{i,i}=0$ for every i), we

see that it is equivalent to the random case. Indeed the constraint $J_{i,i}=0$, when expressed in (20), says that the desired output bit X_μ^1 is now random in regard to the input ones $X_\mu^2, X_\mu^3, \ldots, X_\mu^N$ (X_μ^1 is no longer an input bit, due to the $J_{1,1}=0$ constraint). The difference between L_{easy} and L_{rand} shows that the behaviors of associative memories implemented with the Hebb rule, with or without diagonal elements are not similar; in the first case, when $\alpha\to\infty$, the Hebb solution tends to make all patterns invariant under the dynamics (18) because of the conceptual effect (J tends towards identity), whereas in the second case, the solution becomes random, due to the interferences of the independent learned patterns (as there is only the rote contribution) which leads to confusion.

We remark that $L_{\text{clip}}(\alpha)=L_{\text{rand}}(\pi\alpha/2)$. We can expect from this that the results versus α concerning the associative memories implemented with the Hebb rule without diagonal terms can be similar to its clipped version, modulo the multiplicative transformation $\alpha\to(\pi/2)\alpha$.[13]

VI. CONCLUDING REMARKS

In this paper, we have seen how the Hebb rule (implemented in a perceptron-type architecture) used for the learning of Boolean functions provides two types of contributions. The first one is the rote learning (this contribution depends only weakly on the specified form of the function to be learned). It quickly saturates when the number of learned patterns grows, giving rise to confusion between them.

When the Boolean function to be learned is well suited to the perceptron-type architecture (for example linearly separable, like here) another contribution occurs (the "conceptual" one) which reflects the fact that the inner coherence of the Boolean function is detected and learned by the system. This contribution provides a generalization ability to the system: it not only learns one by one the presented patterns, but also the common coherence between them, when they are numerous enough.

This approach is shown in three simple cases (linearly separable or random) for which analytical results are given. All this is illustrated by numerical examples.

ACKNOWLEDGMENTS

We are very grateful to Ph. Refregier and E. De Chambost for enlightening discussions and to B. Vinter for numerical calculations.

APPENDIX: LEARNING AND GENERALIZATION RATES

We recall here a simple result coming from the central limit theorem, that we will frequently use in this appendix. If g^i's are independent random variables, with the same distribution law, whose mean value is m, and standard deviation σ^2, then

$$\text{Prob}\left[\sum_{i=1}^{q} g^i \geq A\right]=\tfrac{1}{2}\text{erfc}\left[\frac{A-qm}{\sigma\sqrt{2q}}\right], \tag{A1}$$

where we use the complementary error function, defined as follows:

$$\text{erfc}(x)=\frac{2}{\sqrt{\pi}}\int_x^\infty e^{-u^2}du\ . \tag{A2}$$

In this appendix, upper indices refer to components, lower ones to the learning pattern numbering.

1. "Easy case"

For evaluating the generalization rate, we have to calculate the probability that $\sum_{\mu=1}^{\alpha N} z_\mu>0$, where

$$z_\mu=\text{sgn}(\langle B_{\text{easy}}|X_\mu\rangle)\text{sgn}(\langle B_{\text{easy}}|X\rangle)\langle X_\mu|X\rangle \tag{A3}$$

[$|B_{\text{easy}}\rangle=(1,0,0,\ldots,0)$]. We can write

$$z_\mu=X_\mu^1X^1\sum_{j=1}^{N}X_\mu^jX^j\ , \tag{A4}$$

$$z_\mu=1+\sum_{j=2}^{N}X_\mu^1X^1X_\mu^jX^j\ . \tag{A5}$$

The variables $X_\mu^1X^1X_\mu^jX^j$ are independent random variables, equal to 1 or -1 with probability $\frac{1}{2}$. So $\sum_{\mu=1}^{\alpha N} z_\mu$ can be written as

$$\sum_{\mu=1}^{\alpha N} z_\mu=\alpha N+\sum_{i=1}^{\alpha N(N-1)} g^i \tag{A6}$$

where the g^i are random and independent variables, with mean value 0, and standard deviation 1. Using Eq. (A1), we can deduce the generalization rate, in the $N\to\infty$ limit:

$$G_{\text{easy}}(\alpha)=\text{erfc}(-\sqrt{\alpha/2})\ . \tag{A7}$$

In a similar way, with $z_1=N$ and the other z_μ given by (A5), we deduce the learning rate:

$$L_{\text{easy}}(\alpha)=\text{erfc}(-\sqrt{\alpha/2}-\sqrt{1/2\alpha})\ . \tag{A8}$$

2. "Hard case"

We refer to the text for the notations. We begin by evaluating the mean value and the second moment of the random variable z_μ defined as

$$z_\mu=\text{sgn}(\langle B|X\rangle)\langle X|X_\mu\rangle\text{sgn}(\langle B|X_\mu\rangle)\ , \tag{A9}$$

where $|X\rangle$ is fixed (it is the test vector), and $|X_\mu\rangle$ is a random vector on the N-hypercube (its components are independently and randomly chosen equal to ±1, with equal probability $\frac{1}{2}$). We denote the averaging over the $|X_\mu\rangle$'s by $\langle\langle\ \rangle\rangle$. We will then average the result over $|X\rangle$.

We define a linear operator (reducer matrix) $\underline{R}$ from $\mathbb{R}^N$ to $\mathbb{R}^q$ (q is the number of real values defining $|B\rangle$, each one repeated N/q times) by its coordinates:

$$R_{i,j}=\begin{cases}1 & \text{if } (i-1)N/q<j\leq iN/q\\ 0 & \text{otherwise}\ .\end{cases} \tag{A10}$$

For example, if $N=20$ and $q=4$ we have

$$\underline{R}=\begin{bmatrix} 11111 & 00000 & 00000 & 00000 \\ 00000 & 11111 & 00000 & 00000 \\ 00000 & 00000 & 11111 & 00000 \\ 00000 & 00000 & 00000 & 11111 \end{bmatrix} . \tag{A11}$$

Indeed, $\underline{R}(|B\rangle)=(N/q)(B_1,B_2,\ldots,B_q)$. We define another linear operator (duplicator matrix) $\underline{D}$ from $\mathbb{R}^q$ to $\mathbb{R}^N$,

$$\underline{D}=\frac{q}{N}\underline{R}^t . \tag{A12}$$

For example, $\underline{D}\,\underline{R}(|B\rangle)=|B\rangle$, $\underline{R}\,\underline{D}$ is the identity matrix in $\mathbb{R}^q$.

Taking into account symmetry arguments, we can write the following equalities:

$$\langle\langle |X_\mu\rangle \mathrm{sgn}(\langle B|X_\mu\rangle)\rangle\rangle=\underline{D}\,\underline{R}\langle\langle |X_\mu\rangle \mathrm{sgn}(\langle B|X_\mu\rangle)\rangle\rangle , \tag{A13}$$

$$\langle\langle |X_\mu\rangle \mathrm{sgn}(\langle B|X_\mu\rangle)\rangle\rangle=\underline{D}\langle\langle \underline{R}|X_\mu\rangle \mathrm{sgn}(\langle B|\underline{R}^t\underline{R}|X_\mu\rangle)\rangle\rangle . \tag{A14}$$

The principal remark now is that the random vector $\underline{R}|X_\mu\rangle$ has, in the $N\to\infty$ limit, its q components (q is fixed) which tend to have the same Gaussian distribution, centered on the origin, and of standard deviation N/q (central limit theorem). We deduce from this

$$\langle\langle |X_\mu\rangle \mathrm{sgn}(\langle B|X_\mu\rangle)\rangle\rangle=\sqrt{2N/\pi q}\,\underline{D}\frac{\underline{R}|B\rangle}{(\langle B|\underline{R}^t\underline{R}|B\rangle)^{1/2}} , \tag{A15}$$

$$\langle\langle |X_\mu\rangle \mathrm{sgn}(\langle B|X_\mu\rangle)\rangle\rangle=\sqrt{2/\pi}\frac{|B\rangle}{\sqrt{\langle B|B\rangle}} \tag{A16}$$

(we can remark here that this result proves that the Hebb vector tends to have the right direction $|B\rangle$ when $\alpha\to\infty$). One deduces from this

$$\langle\langle z^\mu\rangle\rangle=\mathrm{sgn}(\langle B|X\rangle)\sqrt{2/\pi}\frac{\langle X|B\rangle}{\sqrt{\langle B|B\rangle}} . \tag{A17}$$

It is straightforward to show that

$$\langle\langle (z^\mu)^2\rangle\rangle=N , \tag{A18}$$

and as $\langle\langle z^\mu\rangle\rangle$ is of order 1 ($\langle X|B\rangle$ has a significant probability only when inferior to or of the same order as $\sqrt{N}$, and $\sqrt{\langle B|B\rangle}=\mathrm{const}\sqrt{N}$), the standard deviation of z^μ tends to be equal to $\langle\langle (z^\mu)^2\rangle\rangle=N$. So the generalization rate can be calculated, referring again to (A1),

$$\mathrm{Prob}\left[\sum_{\mu=1}^{\alpha N} z^\mu>0\right] =\tfrac{1}{2}\mathrm{erfc}\left[-\sqrt{\alpha/\pi}\frac{\langle X|B\rangle}{\sqrt{\langle B|B\rangle}}\mathrm{sgn}\langle B|X\rangle\right] . \tag{A19}$$

This expression is the average value, taken over all the possible learning sets (each composed by αN random patterns) of the probability for a given pattern $|X\rangle$ to be recognized well. To get the global generalization rate, one must now average this result over $|X\rangle$.

With similar arguments as the preceding ones, we can show that the random variable u defined by

$$u=\mathrm{sgn}\langle B|X\rangle\frac{\langle X|B\rangle}{\sqrt{\langle B|B\rangle}} \tag{A20}$$

has the following probability law (we remember that $|X\rangle$ is a random test vector on the hypercube):

$$P(u)=\Theta(u)\sqrt{2/\pi}e^{-u^2/2} , \tag{A21}$$

where $\Theta(u)$ is the Heaviside function [$\Theta(u)=1$ if $u>0$, 0 if $u<0$].

We can then deduce the global generalization rate, in the $N\to\infty$ limit:

$$G(\alpha)=\frac{1}{\sqrt{\pi}}\int_0^\infty \mathrm{erfc}(-v\sqrt{2\alpha/\pi})e^{-v^2}dv \tag{A22}$$

($v=u/\sqrt{2}$). In the same way, the learning rate is shown to be

$$L(\alpha)=\frac{1}{\sqrt{\pi}}\int_0^\infty \mathrm{erfc}\left[-\frac{1}{\sqrt{2\alpha}}-v\sqrt{2\alpha/\pi}\right]e^{-v^2}dv . \tag{A23}$$

In fact, the generalization rate can be simplified as follows:[16]

$$G(\alpha)=1-\frac{1}{\pi}\arctan(\sqrt{\pi/2\alpha}) . \tag{A24}$$

3. "Clipping" for the "easy case"

For a test vector $|X\rangle$, we want to evaluate the probability that $X^1\langle X|W_{\mathrm{clip}}\rangle\geq 0$,

$$X^1\langle X|W_{\mathrm{clip}}\rangle=1+\sum_{j=2}^{N}X^1X^j\mathrm{sgn}\left[\sum_{\mu=1}^{\alpha N}X_\mu^1X_\mu^j\right] , \tag{A25}$$

the terms in this summation are independent random values equal to ±1 with equal probability. After Eq. (A1), we deduce

$$G_{\mathrm{rand}}(\alpha)=\tfrac{1}{2}\mathrm{erfc}\left[\frac{-1}{\sqrt{2N}}\right]\to\tfrac{1}{2} . \tag{A26}$$

Concerning the learning rate, the test vector is $|X_1\rangle$:

$$X_1^1\langle X_1|W_{\mathrm{clip}}\rangle=1+\sum_{j=2}^{N}\mathrm{sgn}\left[1+\sum_{\mu=2}^{\alpha N}X_\mu^1X_\mu^jX_1^1X_1^j\right] , \tag{A27}$$

$$X_1^1\langle X_1|W_{\mathrm{clip}}\rangle=1+\sum_{j=2}^{N}g^j , \tag{A28}$$

with straightforward notation for g^j. Referring again to Eq. (A1), we deduce

$$\mathrm{Prob}(g^j=1)=\tfrac{1}{2}\mathrm{erfc}\left[\frac{-1}{\sqrt{2\alpha N}}\right] \tag{A29}$$

that we can simplify by developing $\mathrm{erfc}(x)$ when x is close to 0 ($N\to\infty$):

$$\text{Prob}(g^j=1)\simeq\tfrac{1}{2}[1+\sqrt{2/(\pi\alpha N)}]\ . \tag{A30}$$

So, the mean value of g^j is $\sqrt{2/(\pi\alpha N)}$, and its standard deviation tends towards 1 when $N\to\infty$. Equation (A1) leads then to the following learning rate:

$$L_{\text{rand}}(\alpha)\to\tfrac{1}{2}\text{erfc}\left[\frac{-1}{\sqrt{\pi\alpha}}\right]\ . \tag{A31}$$

4. Random case

In the case of a random Boolean function, only the learning rate is meaningful. For the test pattern $|X_1\rangle$, we have

$$z_\mu=\mathcal{B}(|X_\mu\rangle)\mathcal{B}(|X_1\rangle)\langle X_\mu|X_1\rangle\ , \tag{A32}$$

where $\mathcal{B}$ is the random Boolean function. Equation (A4) now becomes

$$z_\mu=\sum_{j=1}^{N}X_\mu^jX_1^j\mathcal{B}(|X_\mu\rangle)\mathcal{B}(|X_1\rangle)\ , \tag{A33}$$

which is a sum of random variables ±1 for $\mu\geq2$. As $z_1=N$, one deduces the learning rate in the random case, referring again to (A1):

$$L_{\text{rand}}(\alpha)=\text{erfc}(-\sqrt{1/2\alpha})\ . \tag{A34}$$

5. Asymptotic developments

We list the limits when $\alpha\to\infty$:

$$L_{\text{easy}}(\alpha)\sim1-\frac{1}{e\sqrt{2\pi}}\frac{e^{-\alpha/2}}{\sqrt{\alpha}}\ , \tag{A35}$$

$$L_{\text{hard}}(\alpha)\sim1-\frac{1}{\sqrt{2\pi\alpha}}\ , \tag{A36}$$

$$L_{\text{rand}}(\alpha)\sim\tfrac{1}{2}+\frac{1}{\sqrt{2\pi\alpha}}\ , \tag{A37}$$

$$G_{\text{easy}}(\alpha)\sim1-\frac{1}{\sqrt{2\pi}}\frac{e^{-\alpha/2}}{\sqrt{\alpha}}\ , \tag{A38}$$

$$G_{\text{hard}}(\alpha)\sim1-\frac{1}{\sqrt{2\pi\alpha}}\ . \tag{A39}$$

We remark that $G_{\text{hard}}(\alpha)\sim L_{\text{hard}}(\alpha)$.

The limits when $\alpha\to0$ are

$$L_{\text{easy}}(\alpha)\sim1-\frac{1}{e\sqrt{2\pi}}\sqrt{\alpha}e^{-1/(2\alpha)}\ , \tag{A40}$$

$$L_{\text{hard}}(\alpha)\sim1-\frac{e^{1/\pi}\text{erfc}(1/\sqrt{\pi})}{\sqrt{2\pi}}\sqrt{\alpha}e^{-1/(2\alpha)}\ , \tag{A41}$$

$$L_{\text{rand}}(\alpha)\sim1-\frac{1}{\sqrt{2\pi}}\sqrt{\alpha}e^{-1/(2\alpha)}\ , \tag{A42}$$

$$G_{\text{easy}}(\alpha)\sim\tfrac{1}{2}+\sqrt{\alpha/(2\pi)}\ , \tag{A43}$$

$$G_{\text{hard}}(\alpha)\sim\tfrac{1}{2}+\left[\frac{2\alpha}{\pi^3}\right]^{1/2}\ . \tag{A44}$$

[1]DARPA. Neural Network Study. Fairfax-USA, 1988.

[2]H. D. Block, Rev. Mod. Phys. **34**, 123 (1962).

[3]M. Minsky and S. Papert, *Perceptrons* (MIT, Cambridge, 1988).

[4]E. Gardner and B. Derrida, J. Phys. A **22**, 12 (1989).

[5]F. Vallet, Europhys. Lett. **8**, 747 (1989).

[6]E. Gardner, Europhys. Lett. **4**, 481 (1987).

[7]E. Gardner and B. Derrida, J. Phys. A **21**, 271 (1988).

[8]T. Kohonen, *Self Organisation and Associative Memories* (Springer-Verlag, Berlin, 1981).

[9]L. Personnaz, I. Guyon, and G. Dreyfus, J. Phys. Lett. **46**, L359 (1985).

[10]F. Vallet, J.-G. Cailton, and Ph. Réfrégier, Europhys. Lett. **9**, 315 (1989).

[11]W. Krauth and M. Mézard, J. Phys. A **20**, L745 (1987).

[12]E. B. Baum and D. Haussler, Neural Computation **1**, 151 (1989).

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

[14]D. J. Amit, H. Gutfreund, and H. Sompolinsky, Phys. Rev. Lett. **55**, 1530 (1985).

[15]G. Weisbuch and F. Fogelman-Soulié, J. Phys. Lett. **46**, L623 (1985).

[16]*Table of Integrals, Series and Products,* edited by I. S. Gradshteyn and I. M. Ryzhik (Academic, New York, 1980), p. 649, Eq. 6.285.