Storage of structured patterns in a neural network

Patricio Pérez and Dino Salinas
Departamento de Física, Universidad de Santiago de Chile, Casilla 307, Correo 2, Santiago, Chile
(Received 16 March 1994; revised manuscript received 19 July 1994)

In this paper we study the conditions necessary for an autoassociative neural network to store structured patterns built from a predefined set of smaller configurations, which we can treat as words composed by letters. First, we show that no second-order noniterative local learning rule allows an efficient storage of words (but interneural couplings of an order at least equal to the number of letters in a word are neccessary). Besides, for the case of three-letter words, we show that any second-order coupling leads to frustration when two letters are presented. Then, we derive some properties of a neural network model based on a generalized high-order Hebb's rule and coupled subnets, and we show that it solves efficiently the problem of storage and retrieval of words.

PACS number(s): 87.10.+e

INTRODUCTION

Since the tools of statistical physics were first applied to the study of models of associative memory in neural networks [1], particularly making use of techniques such as those used in spin glasses, remarkable advances have been made in the determination of general properties like phase transitions, equilibrium states, and storage capacity [2,3]. These studies have been very complete for neural network models with interactions of any order [4,5] and when storage of noncorrelated patterns is considered. Analytic studies on the storage of correlated patterns have also been performed, but in general the correlation has been very specific, as it is, for example, to assume all patterns with equal magnetization m, which is imposed by choosing them from a biased random distribution [3].

The storage and retrieval of structured patterns, that is, with a correlation defined by the formation of global patterns (or words) from the combination of partial patterns (or letters) stored in subnets, has been studied previously for the case of simple second-order interactions [6,7]. The ubiquity of structured patterns that an organism needs to memorize when facing the detection of patterns of visual, as well as of an auditory nature, or of any other kind, makes it challenging to find out which elements have to be considered in neural network modeling in order to understand this capacity. It is also important to improve learning algorithms that make possible the application of neural networks to a huge amount of practical problems that involve the information management made up by signs, as is the case of texts or other structured signals.

In this paper, we will review the relationship between the number of letters that make up a word and the order of the interneural couplings needed for its memorization. We will also study more extensively a storage algorithm previously presented that combines interactions of second and third order [8].

From now on we will use $h_i(S)$ to represent the value of the local field on the ith neuron ($i=1,\ldots,N$), with the state $S_i=\pm 1$, $S=(S_1,\ldots,S_N)$ being the configuration of the network. The evolution of the network will be given by the asynchronous and random application of the dynamic rule

$$S_i(t+1)=\mathrm{sgn}[h_i(S(t))] \ . \tag{1}$$

In the case of simple interneural couplings or of order 2, J_{ij}, $h_i(S)$ will be assumed to be

$$h_i(S)=\sum_{\substack{j=1\\ j\neq i}}^{N} J_{ij}S_j \ . \tag{2}$$

SIMPLE COUPLINGS AND THEIR LIMITATIONS

A vector $V\in\mathbb{R}^n$ with components $V_i=1$ or -1 will define a letter V in $\mathbb{R}^n$. A set of letters will be called an alphabet. If the letters are generated at random, we would say that the alphabet is random. We define a subnet as the subset of neurons that encodes a letter.

In relation to a subnet x with n neurons, using letters in $\mathbb{R}^n$, the alphabet L_x is defined as

$$L_x=\{L^{1_x},L^{2_x},\ldots,L^{l_x},\ldots,L^{T_x}\} \ , \tag{3}$$

where L^{l_x} is the l_xth letter of the L_x alphabet. For three subnets (1, 2, and 3) of n neurons, the alphabets would be L_1, L_2, and L_3. With these alphabets, it is possible to generate new elements $W^{\nu}\equiv(L^{l_1},L^{l_2},L^{l_3})\in\mathbb{R}^{3n}$ called words, with $l_1=1,\ldots,T_1$, $l_2=1,\ldots,T_2$, and $l_3=1,\ldots,T_3$. The selected words to be stored in a network of $3n$ neurons are clustered in the set

$$P=\{(L^{a_1},L^{a_2},L^{a_3}),(L^{b_1},L^{b_2},L^{b_3}),(L^{c_1},L^{c_2},L^{c_3}),\cdots\} \ , \tag{4}$$

with $a_1\in(1,\ldots,T_1\}$, $a_2\in\{1,\ldots,T_2\}$, etc.

From P we can build a set of word fragments P_{xy} ($x,y=1,2,3$), in such a way that P_{xx} is identically equal to the set of fragments of one letter of the words of P, with such a letter in the L_x alphabet. If $x<y$, P_{xy} is identically equal to the set of two-letter fragments from

words of P, with one letter in the L_x alphabet and the other in the L_y alphabet.

If $x>y$, $P_{xy}\equiv P_{yx}$. So, from P we will have the sets

$$P_{12}=\{(L^{a_1},L^{a_2}),(L^{b_1},L^{b_2}),(L^{c_1},L^{c_2}),\dots\}=P_{21}\ ,$$
$$P_{13}=\{(L^{a_1},L^{a_3}),(L^{b_1}L^{b_3}),(L^{c_1},L^{c_3}),\dots\}=P_{31}\ ,$$
$$P_{23}=\{(L^{a_2},L^{a_3}),(L^{b_2},L^{b_3}),(L^{c_2},L^{c_3}),\dots\}=P_{32}\ ,$$
$$P_{11}=\{L^{a_1},L^{b_1},L^{c_1},\dots\}\ ,$$
$$\vdots \qquad (5)$$

Let i_x and j_y be the i_xth neuron of the x subnet and the j_yth neuron of the y subnet, respectively. Let us define $\{P_{xy}\}$ as the set of all the P_{xy} derived from P. We can build a coupling function $\mathcal{J}_{i_xj_y}$ between the i_x and j_y neurons:

$$\mathcal{J}_{i_xj_y}:\ \{P_{xy}\}\to\mathbb{R}\ , \qquad (6)$$

with $\mathcal{J}_{i_xj_y}(P_{xy})$ equal to the synaptic coupling between the i_x and j_y neurons. We can say that such couplings are of a "regional" class, since they only depend on the components of the stored letters in the x and y subnets. For a local learning rule of a noniterative kind, the synaptic couplings should depend functionally only on the i_x and j_y components of the fragments of words included in P_{xy} (as in the case of Hebb's rule); then they are of the $\mathcal{J}_{i_xj_y}(P_{xy})$ type, though the reciprocal affirmation is not generally true.

Let us suppose that all the couplings between the neurons are of the type

$$J_{i_xj_y}\equiv\mathcal{J}_{i_xj_y}(P_{xy})\ . \qquad (7)$$

The dynamics for the i_x neuron will follow [according to (2)] from the following local field:

$$h_{i_x}(S)=\sum_{\substack{j_x=1\\ j_x\neq i_x}}^{n}\mathcal{J}_{i_xj_x}(P_{xx})S_{j_x}+\sum_{j_y=1}^{n}\mathcal{J}_{i_xj_y}(P_{xy})S_{j_y}$$
$$+\sum_{j_z=1}^{n}\mathcal{J}_{i_xj_z}(P_{xz})S_{j_z}\quad x\neq y,\ \ x\neq z,\ \ y\neq z\ . \qquad (8)$$

From the previous equation, it is evident that, if for different sets P of words we get the same sets P_{xy}, we will have the same dynamics, without the distinction of which set of words has been stored.

Below we will be interested in sets of three-letter words, such that the knowledge of any two letters determines uniquely the complete word (for this to be possible all words must differ in at least two letters). Sets that satisfy this condition will be referred to as "unambiguous sets." The importance of using these sets is that with them it is possible to study the effect of correlation in the retrieval of words from presentations with one missing letter (which we will call "completion of words"). If each of the alphabets has T letters, the largest unambiguous set will have T^2 words. In any of these largest sets, any letter will be part of T words and it will appear with each of the other letters in one word. In what follows, this type of set will be identified as P^k, with k integer. We can build several different sets P^k. (There are at least $T!$ sets, because from any set P^k we can generate a different one through permutations of the letters within one alphabet—that is, interchanging the labeling of letters in Eq. (3). For the same reason, for each set it is possible to find another one of the same type without words in common.) However, the same sets P^k_{xy} come out from these different sets P^k.

In this way, any learning rule of the type mentioned in (7) is not appropriate to store three-letter words. Particularly, any noniterative local rule will not work, as, for example, Hebb's rule. The same reasoning and conclusions could be extended to the storage of p-letter words in relation to the use of couplings of order less than p using the appropriate generalizations of the sets of word fragments (considering fragments with less than p letters) and (7) (considering interneural couplings of order less than p).

Now we will extend our study to the use of any second-order coupling, testing always the completion of stored words. To begin, let us remember that we try to store words denoted as $W^\nu=(L^{l_1},L^{l_2},L^{l_3})$. As mentioned previously, we are interested in the retrieval of a word when we start the dynamics from the same word with one of its letters with 100% noise. In this case the retrieval of the complete word depends on the ability of the neighboring subnets to induce the formation of the correct letter in the noisy subnet (x). Then, for an efficient retrieval of the complete word, it should occur that at $t=0$ the field on any site of the noisy subnet due to the neighboring subnets has the same sign as the letter we expect to retrieve (L^{l_x}). This condition may be expressed as

$$L^{l_x}_{i_x}\left[\sum_{j_y=1}^{n}J_{i_xj_y}L^{l_y}_{j_y}+\sum_{j_z=1}^{n}J_{i_xj_z}L^{l_z}_{j_z}\right]>0\ . \qquad (9)$$

Given that the local field on a given neuron due to activities of and connections with neurons of a neighboring subnet does not depend on what happens on other subnets, it is likely that coupling between subnets has a stabilizing effect for some letters and a destabilizing effect for other letters (given that there are repeated letters). This will lead to frustration.

As an example, let us analyze the following set of unambiguous stored words:

$$P=\{(L^{a_1}L^{a_2}L^{a_3}),(L^{b_1}L^{a_2}L^{b_3}),(L^{b_1}L^{b_2}L^{a_3}),(L^{a_1}L^{b_2}L^{b_3})\}\ . \qquad (10)$$

For the case of the completion of stored words, such that the missing letter be induced in subnet 1, every neuron i_1 should receive the stabilizing influence of the neighboring subnets. If we define the field of subnet y on subnet 1 as $\mathscr{J}_{i_1,y}(L^{l_y})$, from (9) we have

$$\{\mathscr{J}_{i_1,2}(L^{a_2})+\mathscr{J}_{i_1,3}(L^{a_3})\}L_{i_1}^{a_1}>0\ , \tag{11a}$$

$$\{\mathscr{J}_{i_1,2}(L^{a_2})+\mathscr{J}_{i_1,3}(L^{b_3})\}L_{i_1}^{b_1}>0\ , \tag{11b}$$

$$\{\mathscr{J}_{i_1,2}(L^{b_2})+\mathscr{J}_{i_1,3}(L^{a_3})\}L_{i_1}^{b_1}>0\ , \tag{11c}$$

$$\{\mathscr{J}_{i_1,2}(L^{b_2})+\mathscr{J}_{i_1,3}(L^{b_3})\}L_{i_1}^{a_1}>0\ . \tag{11d}$$

Assuming that the letters are randomly generated, $L_{i_1}^{a_1}=-L_{i_1}^{b_1}$ for approximately half of the neurons in a subnet. For these cases, the four inequalities (11) cannot be satisfied simultaneously.

As a verification of the low efficacy of nets with simple couplings in the completion of words, we have performed numerical simulations using an iterative learning algorithm that allows the storage of correlated patterns. We use the Diederich-Opper algorithm [9]. This algorithm guarantees the stability condition

$$L_{i_x}^{l_x}h_{i_x}(W^\nu)=L_{i_x}^{l_x}\left[\sum_{\substack{j_x=1\\ j_x\neq i_x}}^{n}J_{i_xj_x}L_{i_x}^{l_x}+\sum_{j_y=1}^{n}J_{i_xj_y}L_{j_y}^{l_y}+\sum_{j_z=1}^{n}J_{i_xj_z}L_{j_z}^{l_z}\right]\geq\kappa\ , \tag{12}$$

with a bound $\kappa\approx 1$ for r patterns ξ^ν $(\nu=1,\ldots,r)$ with any correlation (for our case $\xi^\nu=W^\nu$). The learning rule is based on an iterative algorithm in which the synaptic coefficients are updated cyclically according to $J_{ij}\rightarrow J_{ij}+\delta J_{ij}$ with

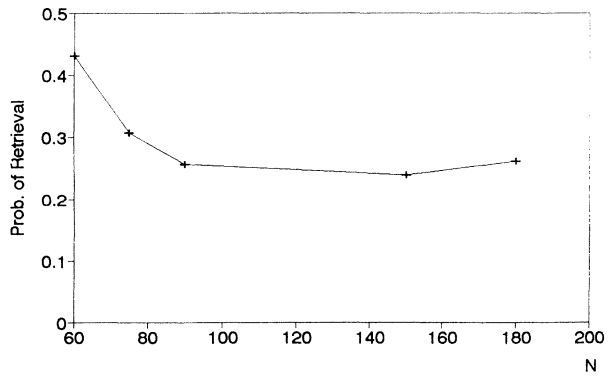

FIG. 1. Probability of retrieval of a three-letter word when we start the dynamics from the same word with the first letter totally noisy. We used nets with a different number of neurons and simple (second-order) interactions generated through the Diederich-Opper's rule.

$$\delta J_{ij}=(N-1)^{-1}\xi_i^\nu\xi_j^\nu\theta(1-h_i(\xi_i^\nu)),\quad j\neq i\ , \tag{13}$$

this being repeated until all patterns satisfy the stability condition. The final couplings will be of the form

$$J_{ij}=(N-1)^{-1}\sum_\nu\chi^\nu\xi_i^\nu\xi_j^\nu\ , \tag{14}$$

where the initial couplings have been chosen to be zero and χ^ν is defined as the number of times the pattern ξ^ν has modified the J_{ij} coupling.

We have carried out a simulation using random three-letter alphabets, which are used to generate a set P^k (previously defined) of words to store. Given that the set is unambiguous, the correct letter to which the noisy subnet should converge is uniquely determined. However, the fraction of events in which the network converged to the appropriate letter was lower than $\frac{1}{3}$ for n large enough (Fig. 1). In all cases the noiseless subnets kept their states fixed during the dynamics.

USE OF HIGH-ORDER INTERACTIONS FOR THE STORAGE OF THREE-LETTER WORDS

Now, in relation to the storage of three-letter words generated with random alphabets, we will study a noniterative local algorithm. This is based upon the generalized Hebb rule, and it considers the coupling through interactions of order 3 between neurons of different n-neuron subnets $(N=3n)$ and interactions of order 2 for neurons within a subnet.

The dynamics for neuron i_x will follow from the following local field:

$$h_{i_x}(S)=\sum_{\substack{j_x=1\\ j_x\neq i_x}}^{n}J_{i_xj_x}S_{j_x}+\lambda\sum_{\substack{j_y=1\\ j_z=1}}^{n}J_{i_xj_yj_z}S_{j_y}S_{j_z}\ ,\qquad x\neq y,\ x\neq z,\ y\neq z\ , \tag{15}$$

with λ a positive parameter that modulates the intensity of couplings, which are chosen according to a generalization of Hebb's rule and as a function of the selected words W^ν to be stored:

$$J_{i_xj_x}=\frac{1}{N}\sum_\nu W_{i_x}^\nu W_{j_x}^\nu\ , \tag{16a}$$

$$J_{i_xj_yj_z}=\frac{3}{N^2}\sum_\nu W_{i_x}^\nu W_{j_y}^\nu W_{j_z}^\nu\ . \tag{16b}$$

Let us define $\lambda_{l_1l_2l_3}$ as an indicator of the absence (0) or presence (1) of the word $W^\nu\equiv(L^{l_1},L^{l_2},L^{l_3})$. Then, the final expression for Eq. (15) is

$$h_{i_x}(S)=\frac{1}{3n}\sum_{\substack{j_x=1\\ j_x\neq i_x}}^{n}\sum_{l_xl_yl_z}^{T}\lambda_{l_1l_2l_3}L_{i_x}^{l_x}L_{j_x}^{l_x}S_{j_x}+\frac{\lambda}{3n^2}\sum_{\substack{j_y=1\\ j_z=1}}^{n}\sum_{l_xl_yl_z}^{T}\lambda_{l_1l_2l_3}L_{i_x}^{l_x}L_{j_y}^{l_y}L_{j_z}^{l_z}S_{j_y}S_{j_z}\ . \tag{17}$$

We are interested in the capacity to retrieve a word

when the dynamics is started with one of its letters unrecognizable and the other two letters correct (which should determine uniquely the missing one), so we will store again unambiguous sets P^k. Then, Eq. (17) can be written more explicitly, considering that each letter appears in T words:

$$h_{i_x}(S)=\frac{T}{3n}\sum_{\substack{j_x=1\\ j_x\neq i_x}}^{n}\sum_{l_x}^{T}L_{i_x}^{l_x}L_{j_x}^{l_x}S_{j_x}+\frac{\lambda}{3n^2}\sum_{\substack{j_y=1\\ j_z=1}}^{n}\sum_{l_x l_y l_z}^{T}\lambda_{l_1 l_2 l_3}L_{i_x}^{l_x}L_{j_y}^{l_y}L_{j_z}^{l_z}S_{j_y}S_{j_z}\ . \tag{18}$$

Let us assume that the state S of the network is one of the selected words $W^\mu\equiv(L^{a_1},L^{a_2},L^{a_3})$. In that case the local field $h_{i_x}(W^\mu)$ can be decomposed in a noise term, which tends to make the word unstable, and a signal term, which tends to reinforce it. For $n\gg T$, the noise term has a Gaussian distribution of zero mean and a standard deviation

$$\sigma_R=\left[\frac{(T-1)}{9n}(T^2+2\lambda^2)\right]^{1/2}\ . \tag{19}$$

The signal term is

$$I=L_{i_x}^{a_x}(T+\lambda)\tfrac{1}{3}\ . \tag{20}$$

For any nonstored word it will be possible to find one of the stored ones with which there is a difference of only one letter. Let us look to a word that has a unique letter (L^{b_x}) different from the previous word. The local field for this nonstored word will have a signal term given by

$$L_{i_x}^{b_x}(T+\lambda L_{i_x}^{b_x}L_{i_x}^{a_x})\tfrac{1}{3}\ . \tag{21}$$

For those sites where both letters are not coincident, the signal term will destabilize the nondesirable word when $\lambda>T$.

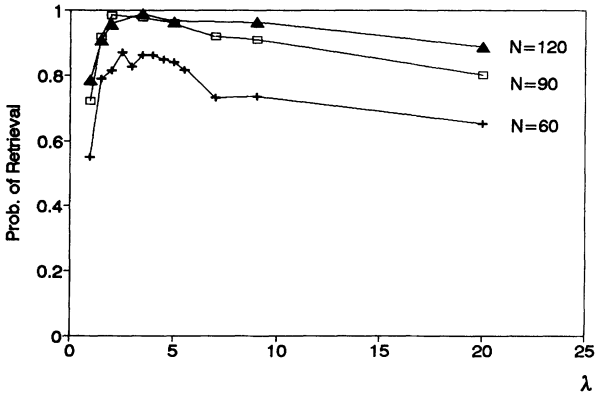

FIG. 2. Probability of retrieval of a word with a noisy letter as a function of the parameter of intersubnet coupling for nets with a different total number of neurons and three letter alphabets. The model is based on second-order intrasubnet coupling and third-order intersubnet coupling.

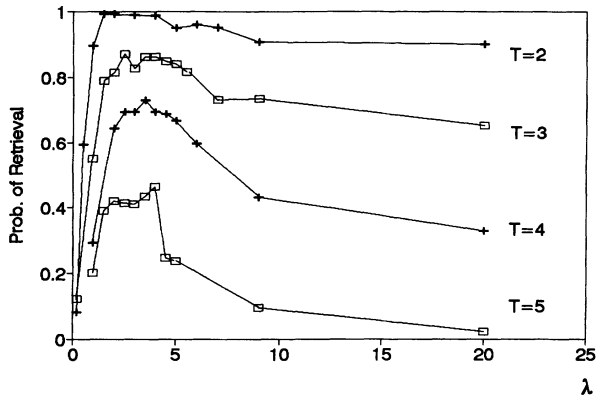

FIG. 3. Probability of retrieval of a word with a noisy letter as a function of the parameter of intersubnet coupling for 60 neuron nets. Each curve refers to the use of alphabets with a different number T of letters. T^2 words, which differ in at least two letters, are stored. The model is the same as in Fig. 2.

In order for the word W^μ to be stored, the condition of local stability must be satisfied:

$$h_{i_x}(W^\mu)L_{i_x}^{a_x}>0\ , \tag{22}$$

for all neurons in the net.

The probability Q that, after generating the alphabets randomly, a word $W^\mu\in P^k$ satisfies (22) is (neglecting correlation between sites)

$$Q=\left[\int_{-|I|/\sigma_R}^{\infty}dx\frac{1}{\sqrt{2\pi}}e^{-x^2/2}\right]^N\ . \tag{23}$$

Using (19) and (20), it can be shown that Q increases with N, decreases with T, and that it has a single maximum at $\lambda=T/2$.

We have simulated N-neuron networks clustered in three subnets of n neurons each, with the intention of storing a set P^k using T-letter alphabets. We could calculate the probability to retrieve a word from a P^k when one of its letters had 100% noise (completion of words), for different values of λ. The results are shown in Figs. 2 and 3: The curves show an increase in the probability for increasing N, a decrease with increasing T, and maxima in $\lambda_{\max}=O(T)$, not far from the value $T/2$ predicted above.

DISCUSSION

We have proved the necessity of including neural interactions of an order higher or equal to the number of letters of words to be stored when using noniterative local memorization rules, whether the alphabets are random or not. The central idea was to fragment the selected words and to build sets of equal size fragments. If the synaptic couplings are functions of a set of fragments, when there is more than one set of words that generate that variety of fragments, the network loses its selective storage capacity. It is easy to verify that for each set of words defined

as unambiguous, there exists another set of the same size with no words in common that generates the same fragments.

Besides, we have analyzed the ability of a net on a task defined as completion of a stored word. Here we were based upon the idea that we wanted to retrieve a whole word starting from the same incomplete word with only one unknown letter. For three-letter words, we have realized that there is no learning rule with second-order coupling that can be efficient in the completion of words from any unambiguous stored set. The reason for this is that the field from the neighboring subnets on the noisy subnet may be decomposed into two independent terms, such that the contribution of any of them to the retrieval of the noisy letter does not depend on the combination of letters present. Then, frustration may arise as a result of the topology of given groups of stored words [Eq. (10)], and will not depend on the specific model of second-order couplings used [Eqs. (11)]. A similar reasoning may be applied to the general case of the storage of p-letter words. In this case, in order to have efficient storage, at least couplings of order p should be used.

Our conclusions are not in contradiction with the results obtained by Krey and Pöppel [6], who use second-order couplings to store words with an arbitrary number of letters, because they have not considered the storage of words with repeated letters.. Besides, their more detailed analysis is restricted to the case of two-letter words.

The method of coupled subnets presented here (which can be generalized to the storage of words with more than three letters) allows for a decrease in the computational cost when including simple interactions instead of higher-order interactions within the subnets, without decreasing the efficiency of the net (as we have observed). Moreover, with the use of this generalized Hebb rule it is easy to store a new word, linearly modifying the couplings.

We may ask if a model based on a multilayer perceptron, which uses only second-order couplings, could be more efficient than an autoassociative network in the storage and retrieval of words. Studies in this direction are underway.

ACKNOWLEDGMENTS

We would like to thank the Fondo Nacional de Ciencia y Tecnología (FONDECYT) (Project No. 1930106) and Departamento de Investigaciones Científicas y Tecnológicas, Universidada de Santiago (DICYT) for their support.

[1] J. J. Hopfield, Proc. Natl. Acad. Sci. U.S.A. **79**, 2554 (1982).
[2] D. J. Amit, H. Gutfreund, and H. Sompolinsky, Ann. Phys. **173**, 30 (1987).
[3] E. Gardner, J. Phys. A **21**, 257 (1988).
[4] E. Gardner, J. Phys. A **20**, 3453 (1987).
[5] G. A. Kohring, J. Phys. (Paris) **51**, 145 (1990).
[6] U. Krey and G. Pöppel, Z. Phys. B **76**, 513 (1989).
[7] E. Gardner, N. Stroud, and D. J. Wallace, J. Phys. A **22**, 2019 (1989).
[8] P. Pérez and G. Salini, Phys. Lett. A **181**, 61 (1993).
[9] S. Diederich and M. Opper, Phys. Rev. Lett. **58**, 949 (1987).