OF POINTS AND LOOPS

I. Guyon, L. Personnaz, G. Dreyfus
Ecole Supérieure de Physique et de Chimie Industrielles
de la Ville de Paris
Laboratoire d'Electronique
10, rue Vauquelin
F - 75005 Paris
France

ABSTRACT

New learning rules for the storage and retrieval of temporal sequences, in neural networks with parallel synchronous dynamics, are presented. They allow either one-shot, non-local learning, or slow, local learning. Sequences with bifurcation points, i.e. sequences in which a given state appears twice, or in which a given state belongs to two distinct sequences, can be stored without errors and retrieved.

1. INTRODUCTION

The original attempts to use Hopfield-type neural networks as associative memories aimed at storing information as fixed points of the dynamics of the systems. However, many pieces of information appear naturally as *temporal sequences* : speech, music, flow charts, etc. Obviously, the central nervous system has the ability to store, retrieve and recognize sequences of patterns. Therefore, various attempts have been made recently in this direction. Most of them, however, were aimed at biological modelling : they used networks with *sequential asynchronous* dynamics, in which the main problem is the competition between the stability of a pattern and the transition to the next one ; in the present paper, we study the behaviour of neural networks with *parallel synchronous* operation. We show that it is possible to find efficient learning rules which allow the perfect storage, and the retrieval, of complex sequences, i.e. of sequences in which a given pattern occurs more than once. Some of the results which are obtained apply to asynchronous dynamics, too. In the first section, we shall recall the various learning

NATO ASI Series, Vol. F41
Neural Computers
Edited by R. Eckmiller and Ch. v. d. Malsburg

processes which have been proposed and used for storing patterns as fixed points. In a subsequent section, we shall present various learning rules for temporal sequences.

2. THE STRUCTURE AND DYNAMICS OF THE NETWORKS.

The neural networks considered here are assemblies of McCulloch-Pitts binary formal neurons, having the following operation : each neuron computes its potential, which is the weighted sum of its inputs, and makes a decision by comparing it to a predetermined threshold; if the sum is larger than the threshold, the neuron goes to (or remains in) the active state ; if the sum is smaller than the threshold, the neuron goes to (or remains in) the inactive state. We denote the state of neuron i by a variable σ_i which can take the values +1 or -1 only ; C_{ij} is the weight of the synapse inputting information from neuron j to neuron i ; we shall take all thresholds equal to zero. Such neurons are arranged to form a fully connected network. Therefore, the evolution of the state of neuron i is governed by the following process : denoting by v_i the potential of neuron i in a network of n neurons

$$v_i(t) = \sum_{j=1}^{n} C_{ij}\, \sigma_j(t) \quad \text{, one has :}$$

$$\sigma_i\, (t+\tau) = \text{sign}\, (v_i(t)) \,.$$

where τ is the characteristic response time of the neuron.

Unless otherwise stated, we consider neural networks with parallel, synchronous dynamics : all neurons evaluate their potentials and make their decisions simultaneously, with the same response time τ. The state of a network of n neurons is represented by a vector $\underline{\sigma}$ whose n components are equal to ± 1. The points in state space that can be occupied by the network are the summits of a hypercube. The dynamics of the network is fully defined by the values of the interaction coefficients C_{ij} ; the matrix C of these coefficients is usually termed the synaptic matrix of the system.

3. STORING PATTERNS AS FIXED POINTS

Hopfield-type networks[1], as opposed to feedforward networks, are essentially dynamical systems ; classically, their use as associative memories is based on the fact that information can be stored as attractor, fixed points of their dynamics : when left to evolve spontaneously from an initial state, which corresponds to an erroneous or incomplete information, the network converges to a fixed point which is, hopefully, the correct information ; of course, this property holds true only if the interaction coefficients are properly computed during the learning phase.

The first learning rule guaranteeing the perfect storage of any set of patterns was proposed in Ref. 2. We summarize briefly its derivation : the stability of a state $\underline{\sigma}^k$ is guaranteed iff one has :

$$C\,\underline{\sigma}^k = A^k\,\underline{\sigma}^k = \underline{\lambda}^k \qquad (1),$$

where A^k is any diagonally dominant matrix with positive diagonal terms. Therefore, any synaptic matrix guaranteeing the stability of a set of states $\{\underline{\sigma}^k\}$ must satisfy relation (1) for all k=1, 2, ... p ; the general solution is given by:

$$C = \Lambda\,\Sigma^I + B\,(I - \Sigma\,\Sigma^I) \qquad (2),$$

where Λ is the matrix whose columns are the vectors $\underline{\lambda}^k$

$$\Lambda = [\,\underline{\lambda}^1,\ \underline{\lambda}^2, \ldots\ldots,\ \underline{\lambda}^p\,],$$
$$\Sigma = [\,\underline{\sigma}^1,\ \underline{\sigma}^2, \ldots\ldots,\ \underline{\sigma}^p\,],$$

B is an arbitrary matrix and Σ^I is the pseudoinverse of Σ. This holds true provided one has $\Lambda\,\Sigma^I\,\Sigma = \Lambda.$ (3)

The computation of the synaptic matrix can be performed iteratively, by presenting each pattern to be learnt only once ; this learning procedure will be detailed in the next section. Note that matrix B is the synaptic matrix at the beginning of the learning phase, i.e. when $\Sigma = [0]$.

If condition (3) is not satisfied, matrix $C = \Lambda\,\Sigma^I$ minimizes the quantity

$$\sum_{i=1}^{n}\sum_{k=1}^{p}(\,v_i^k - \lambda_i^k\,)^2 \quad .$$

A particularly simple and important result is obtained if $\Lambda = \Sigma$, i.e., if all A^k are taken equal to the identity matrix : in this case, if learning begins with a *tabula rasa* (B=[0]), the synaptic matrix reduces to the orthogonal projection matrix into the subspace spanned by the stored patterns. This gives rise to a very simple geometrical interpretation of the information retrieval properties of the networks, and allows us to define a Lyapunov function for the study of the parallel dynamics of the system ; moreover, it turns out to be very efficient for applications in the field of pattern recognition. This learning rule, termed the projection rule, has been analyzed in detail in Ref. 3 , and, in a slightly different form, in Ref. 4 ; applications to pattern recognition are described in Ref. 5 and 6.

The above learning procedure is *fast* and *non-local*. It is fast because the iterative computation of the synaptic matrix requires presenting each pattern to be learnt only once ("one-shot learning") ; if p patterns are to be stored, p iterations will be necessary. This rule is non-local since computing the variation of the strength of the synapse linking neurons i and j requires information from all other neurons of the network ; from a practical standpoint, it would be more advantageous to have a local learning rule, which would make the implementation of the learning rule on an electronic or electro-optic neural network much easier. The price to be paid for a local rule is that the learning procedure is slow since it requires that each pattern be presented repeatedly. Two such procedures have been proposed recently[7]. One of them is derived from the Widrow-Hoff rule[8] and has been shown to yield the projection matrix if the stored patterns are linearly independent. The other local learning procedure is a variant of the Perceptron rule[9] ; it guarantees the stability of the stored patterns, but the synaptic matrix is of the general form (2) ; it is not the projection matrix ; the maximal storage capacity is 2n if the patterns are chosen randomly[10].

4. STORAGE AND RETRIEVAL OF SEQUENCES OF PATTERNS

The Hopfield network, being essentially a dynamical system, is an attractive candidate for processing temporal sequences. Recently, several authors proposed network architectures and learning rules for storage, retrieval and/or recognition of sequences[11], essentially in the framework of sequential dynamics. The problem that we address here is the storage and retrieval of sequences with neural networks under parallel synchronous dynamics. We shall first recall results obtained previously for the storage of simple sequences, and subsequently show how they can be extended to complex sequences, leading to learning rules which guarantee the perfect storage of temporal sequences.

1) Storage and retrieval of simple sequences

Storing a sequence consists in computing the synaptic coefficients so as to impose a set of prescribed one-step transitions in state space. Consider a set of transitions :

$$\underline{\sigma}^k \rightarrow \underline{\sigma}^{k+} \quad , k=1, 2, ..., p \ .$$

We wish to compute the synaptic matrix in order to guarantee that, if the network is in state $\underline{\sigma}^k$, it is in state $\underline{\sigma}^{k+}$ at the next time step. In other words, we wish to

impose the condition : $C\,\underline{\sigma}^k = A^k\,\underline{\sigma}^{k+} = \underline{\lambda}^k$ for all k,
where A^k is an arbitrary diagonally dominant matrix with positive diagonal terms. As mentioned in the previous section, the general solution of this equation is :

$$C = \Lambda\,\Sigma^I + B\,(\,I - \Sigma\,\Sigma^I\,)\,.$$

If we impose the same margin for all neurons and all transitions, that is, if we take $A^k = I$ for all k, one has :

$$C = \Sigma^+\,\Sigma^I + B\,(\,I - \Sigma\,\Sigma^I\,)\,,$$

provided that condition $\Sigma^+\,\Sigma^I\,\Sigma = \Sigma$ is satisfied.

With this learning rule, the number of transitions that can be stored is O(n), with the restriction that a given pattern must not appear twice in matrix Σ. Figure 1 illustrates the type of sequences that can be stored with this rule. Examples of applications to classification tasks performed with such networks are presented in Reference 3.

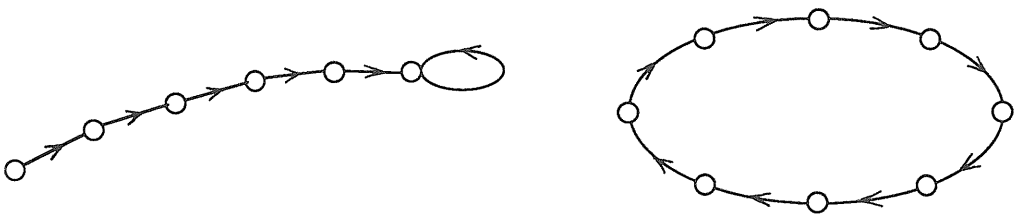

Figure 1

As in the case of pattern storage, matrix C can be computed iteratively with single presentation of the transitions to be stored. Starting from a zero synaptic matrix, the computation of $C = \Sigma^+\,\Sigma^I$
proceeds as follows : assume that k-1 elementary transitions

$$\underline{\sigma}^h \rightarrow \underline{\sigma}^{h+} \quad (h=1, 2, ..., k-1)\,,$$

have been learnt, leading to a synaptic matrix C(k-1). Then, C(k) can be computed as :

$$C(k) = C(k-1) + (\,\underline{\sigma}^{k+} - \underline{v}^k\,)\,\tilde{\underline{\sigma}}^k \,/\, \|\,\tilde{\underline{\sigma}}^k\|^2 \qquad (4)$$

where $\underline{v}^k = C(k-1)\,\underline{\sigma}^k$, $\quad \tilde{\underline{\sigma}}^k = M(k-1)\,\underline{\sigma}^k$

and $\quad M(k) = M(k-1) - \tilde{\underline{\sigma}}^k\,\tilde{\underline{\sigma}}^{kT} \,/\, \|\,\tilde{\underline{\sigma}}^k\|^2$

the initial conditions being

$C(0) = [0]$, $M(0) = I$.

This fast learning algorithm is non-local because of the term $\tilde{\underline{\sigma}}^k$ in the relation (4). Local learning can be achieved by a straightforward generalization of the slow-learning procedures (Widrow-Hoff or Perceptron type) developed for storing patterns as fixed points. The Widrow-Hoff algorithm can be written as :

$$C(k) = C(k\text{-}1) + (1/n)\ (\underline{\sigma}^{k+} - \underline{v}^{k})\ \underline{\sigma}^{kT} \qquad (5)$$

with $C(0) = [0]$.

The Perceptron-type procedures allow us to find a synaptic matrix whose coefficients satisfy a set of inequalities :

$$\Sigma_j\ C_{ij}\ \sigma_j^{k}\ \sigma_i^{k+} > \delta > 0\ .$$

If δ is smaller than a limiting value δ_{max} which can be determined[12], a solution exists and will be found by the Perceptron algorithm in a finite number of steps.

2) Storage and retrieval of complex sequences

In order to learn sequences in which a given pattern occurs twice, the previous approach is inappropriate : consider a pattern $\underline{\sigma}^k$ belonging to two distinct sequences, or appearing twice in the same sequence (Fig. 2) ; when the network is in that state, it must decide which of the subsequent possible states it must go to at the next step. Therefore, in order to make such a decision, some information on the previous state must be conveyed ; this is not possible with the structure described in the previous section. A solution to this problem consists in performing, at each neuron, two weighted sums, one of them taking into account the present state of the network, the other involving its previous state.

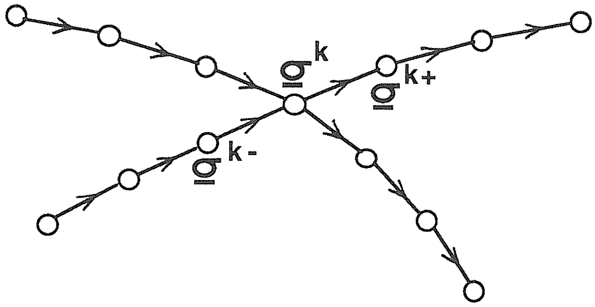

Figure 2

More generally, we define the order of a set of sequences as the minimal memory span necessary to store all the sequences. Hence, a sequence where a pattern occurs twice is of order one. Simple sequences, discussed in the previous section, are of order zero. In the following, we consider sequences of order one, but extensions to higher order are straightforward. Two possible solutions, for sequences of order one, will be discussed here.

As mentioned above, the network needs information both on the current state and on the previous one ; we denote $\underline{\sigma}(t-1)$, $\underline{\sigma}(t)$ and $\underline{\sigma}(t+1)$ by $\underline{\sigma}^-$, $\underline{\sigma}$ and $\underline{\sigma}^+$ respectively. The dynamics of the network is now governed by the following relation :

$$\underline{\sigma}^+ = \text{sign}\,[\,C\,\underline{\gamma}\,]\ ,$$

where $\underline{\gamma}$ is a vector with 2n components :

$$\underline{\gamma} = \begin{bmatrix} \underline{\sigma}^- \\ \underline{\sigma} \end{bmatrix}$$

Therefore, C is a (n,2n) matrix.

The computation of C proceeds as follows : the sequences to be stored can be put in the form

$$\underline{\sigma}^{k-} \rightarrow \underline{\sigma}^{k} \rightarrow \underline{\sigma}^{k+} \quad k=1, ..., p.$$

We define a matrix Γ whose columns are the vectors $\underline{\gamma}^k$. The usual storage condition for the sequences can be written as :

$$C\,\underline{\gamma}^k = A^k\,\underline{\sigma}^k\ .$$

Taking all A^k equal to the identity matrix, and starting from an initially unconnected network, matrix C is given by :

$$C = \Sigma^+\,\Gamma^I,$$

under the condition $\Sigma^+\,\Gamma^I\,\Gamma = \Sigma^+$.

This structure of the network guarantees the storage of any sequence of order 1. It requires $2n^2$ synaptic coefficients, and the number of transitions that can be stored is O(2n). The retrieval requires that the network be initialized with two states.

An alternate solution can be used, allowing the initialization of the process with only one pattern. In this case, we decompose C into two square (n,n) matrices $C^{(0)}$ and $C^{(1)}$, $C^{(0)}$ acting on $\underline{\sigma}$ and $C^{(1)}$ on $\underline{\sigma}^-$. Therefore, the dynamics for the network is :

$$\underline{\sigma}^+ = \text{sign}\,(\,C^{(0)}\,\underline{\sigma} + C^{(1)}\,\underline{\sigma}^-\,)\ .$$

The sequences to be stored are divided into subsequences :

$$\underline{\sigma}^k \rightarrow \underline{\sigma}^{k+} \rightarrow \underline{\sigma}^{k++} \quad k=1, ..., p\ .$$

The storage of the sequences is guaranteed if :

$$C^{(0)}\,\underline{\sigma}^k = \underline{\sigma}^{k+} \quad \text{and} \quad C^{(1)}\,\underline{\sigma}^k = \underline{\sigma}^{k++} \quad k=1, ..., p\ ,$$

which can be put in matrix form as :

$$C^{(0)}\,\Sigma = \Sigma^+ \quad \text{and}\ C^{(1)}\,\Sigma = \Sigma^{++}\ .$$

As opposed to the previous case, the solutions

$$C^{(0)} = \Sigma^{+}\,\Sigma^{I} \quad \text{and} \quad C^{(1)} = \Sigma^{++}\,\Sigma^{I}$$

are not exact solutions because the bifurcation vectors appear twice in Σ. This difficulty can be overcome in the following way : we define matrix S as the matrix derived from matrix Σ by deleting the columns of Σ corresponding to all occurrences, but the first one, of each bifurcation point ; matrices S^{+} and S^{++} are derived from Σ^{+} and Σ^{++} by deleting the same columns as for S, and by replacing the columns corresponding to the bifurcation points by $\underline{0}$. Synaptic matrices guaranteeing the storage of the sequences are given by :

$$C^{(0)} = S^{+}\,S^{I} \quad \text{and} \quad C^{(1)} = S^{++}\,S^{I}.$$

When this solution is used, the network can be initialized with one state only, the second state being equal to $\underline{0}$. The storage capacity, however, is decreased by a factor of 2 as compared to the previous solution, and one cannot have two bifurcation points in succession in a sequence.

With either solutions, straightforward extensions of the local rules presented are available and allow slow, local learning. Other solutions to the problem of sequence learning and retrieval, together with some illustrations, will be presented in a more detailed paper[13].

5. CONCLUSION

The present paper has introduced new storage prescriptions which guarantee the storage, and allow the subsequent retrieval, of temporal sequences of information in networks with parallel synchronous dynamics. Having in view the possible electronic implementations of such systems, we have shown that local learning rules can be used, in addition to the fast, non-local learning rules which are more suitable for off-chip computation of the synaptic matrix. The ability to store and retrieve information in the form of temporal sequences extends the range of tools which are available to the "neural network designer" for attempting to solve information processing problems with such systems.

Acknowledgments : the authors are grateful to J.P. Nadal for stimulating discussions.

REFERENCES

1. J. J. Hopfield, Proc. Natl. Acad. Sci. (USA) **79**, 2554 (1982).
2. L. Personnaz, I. Guyon, G. Dreyfus, J. Phys. Lett. **46**, L359 (1985)
3. L. Personnaz, I. Guyon, G. Dreyfus, Phys. Rev. A **34,** 4217 (1986).
4. I. Kanter, H. Sompolinsky, Phys. Rev. A **35**, 380 (1987).
5. L. Personnaz, I. Guyon, P. Siarry, G. Dreyfus, *Heidelberg Colloquium on Glassy Dynamics and Optimization* (Springer, 1987).
6. L. Personnaz, I. Guyon, G. Dreyfus, *Computational Systems - Natural and Artificial* (Springer, 1988).
7. S. Diederich, M. Opper, Phys. Rev. Lett. **58**, 949 (1987).
8. B. Widrow, M.E. Hoff, *IRE Wescon Conv. Record* , **4**, 96 (1960).
9. F. Rosenblatt, *Principles of Neurodynamics,* Spartan Books, N.Y. 1962.
10. E. Gardner, preprint (1987)
11. D. W. Tank, J. J. Hopfield, preprint 1986.
    H. Sompolinsky, I. Kanter, Phys. Rev. Lett. **57**, 2861 (1986).
    D. Kleinfeld, PNAS (USA) **83**, 9469 (1987).
    S. Dehaene, J. P. Changeux, J. P. Nadal, PNAS (USA) **84**, 2727 (1987).
    J. Buhmann, K. Schulten, preprint 1987.
    D. Kleinfeld, H. Sompolinsky, preprint 1987.
    D. Amit, preprint 1987.
12. W. Kraunt, M. Mézard, preprint 1987.
13. L. Personnaz, I. Guyon, G. Dreyfus, J.P. Nadal, preprint 1987.