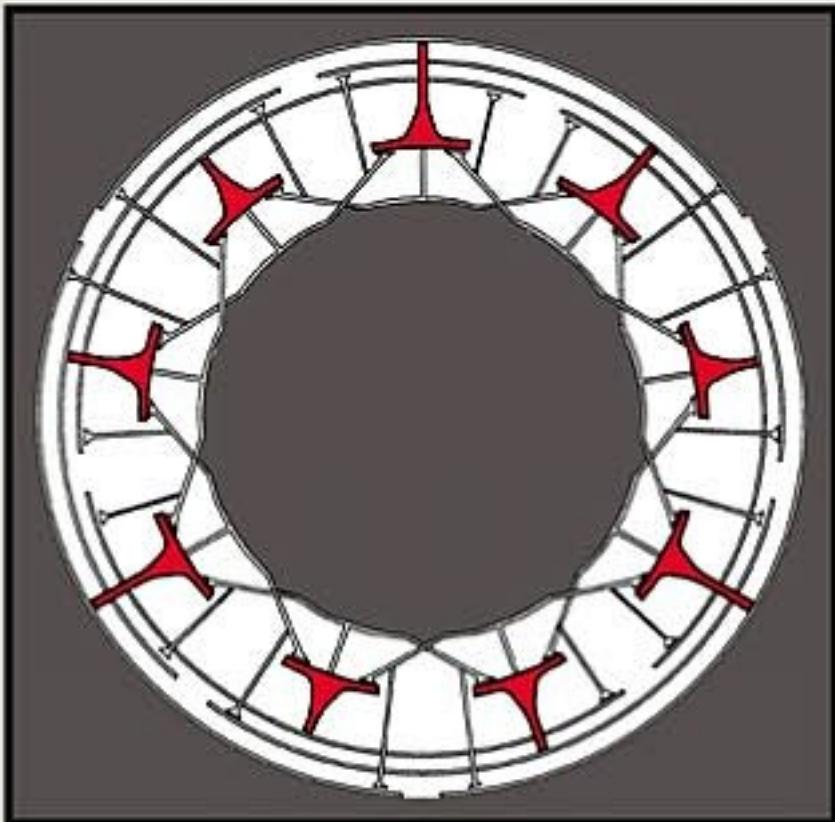


P. Peretto

An Introduction to the Modeling of Neural Networks



Collection
*Aléa
Saclay*

This text is a beginning graduate-level introduction to neural networks, focussing on current theoretical models, examining what these models can reveal about the brain functions, and discussing the ramifications for psychology, artificial intelligence and the construction of a new generation of intelligent computers.

The book is divided into four parts. The first part gives an account of the anatomy of the central nervous system, followed by a brief introduction to neurophysiology. The second part is devoted to the dynamics of neuronal states, and demonstrates how very simple models may simulate associative memory. The third part of the book discusses models of learning, including detailed discussions on the limits of memory storage, methods of learning and their associated models, associativity, and error correction. The final part reviews possible applications of neural networks in artificial intelligence, expert systems, optimization problems, and the construction of actual neural supercomputers, with the potential for one-hundred-fold increase in speed over contemporary serial machines.

COLLECTION ALÉA-SACLAY: Monographs and Texts in Statistical Physics 2

GENERAL EDITOR: C. Godrèche

**AN INTRODUCTION TO THE MODELING OF
NEURAL NETWORKS**

COLLECTION ALÉA-SACLAY: Monographs and Texts in Statistical Physics

- 1 C. GODRÈCHE (ed): Solids far from Equilibrium**
- 2 P. PERETTO: An Introduction to the Modeling of Neural Networks**

AN INTRODUCTION TO THE MODELING OF NEURAL NETWORKS

PIERRE PERETTO

Centre d'Études de Grenoble



Published by the Press Syndicate of the University of Cambridge
The Pitt Building, Trumpington Street, Cambridge CB2 1RP
40 West 20th Street, New York, NY 10011-4211, USA
10 Stamford Road, Oakleigh, Melbourne 3166, Australia

© Cambridge University Press 1992

First published 1992
Reprinted 1994

A catalogue record for this book is available from the British Library

Library of Congress cataloguing in publication data available

ISBN 0 521 41451 2 hardback
ISBN 0 521 42487 9 paperback

Transferred to digital printing 2004

To my wife, Michèle, and my children

CONTENTS

Preface	xxxiii
Acknowledgments	xvii
1 Introduction	1
1.1 Mind as an emergent property of nervous systems	1
1.2 Neuronal nets as automata networks: a brief historical overview	6
1.3 Organization of the book	11
2 The biology of neural networks: a few features for the sake of non-biologists	13
2.1 Three approaches to the study of the functioning of central nervous systems	13
2.2 The anatomy of central nervous systems	15
2.3 A brief survey of neurophysiology	29
2.4 Learning and memory: a summary of experimental observations	41
3 The dynamics of neural networks: a stochastic approach	57
3.1 Introducing the problem	57
3.2 Noiseless neural networks	64
3.3 Taking synaptic noise into account	78

4 Hebbian models of associative memory	99
4.1 Noiseless Hebbian models	99
4.2 Stochastic Hebbian neural networks in the limit of finite numbers of memorized patterns	112
4.3 Storing an infinite number of patterns in stochastic Hebbian networks: the technique of field distributions	130
4.4 The replica method approach	141
4.5 General dynamics of neural networks	149
5 Temporal sequences of patterns	153
5.1 Parallel dynamics	153
5.2 Stochastic dynamics	155
5.3 An example of conditioned behavior	168
6 The problem of learning in neural networks	173
6.1 Introducing the problem	173
6.2 Linear separability	183
6.3 Computing the volume of solutions	196
7 Learning dynamics in ‘visible’ neural networks	209
7.1 A classification of learning dynamics	209
7.2 Constraining the synaptic efficacies	212
7.3 Projection algorithms	218
7.4 The perceptron learning rules	230
7.5 Correlated patterns	248
8 Solving the problem of credit assignment	269
8.1 The back-propagation algorithm	269
8.2 Handling internal representations	278
8.3 Learning in Boolean networks	292
9 Self-organization	299
9.1 Self-organization in simple networks	299
9.2 Ontogenesis	307
9.3 Three questions about learning	319

10 Neurocomputation	325
10.1 Domains of applications of neural networks	325
10.2 Optimization	326
10.3 Low-level signal processing	350
10.4 Pattern matching	357
10.5 Some speculations on biological systems	366
10.6 Higher associative functions	371
11 Neurocomputers	379
11.1 General principles of neurocomputation	379
11.2 Semi-parallel neurocomputers	391
12 A critical view of the modeling of neural networks	403
12.1 Information structures the biological system	403
12.2 The neural code	404
12.3 The synfire chains	405
12.4 Computing with attractors versus computing with flows of information	406
12.5 The issue of low neuronal activities	408
12.6 Learning and cortical plasticity	413
12.7 Taking the modular organization of the cortex into account	414
12.8 Higher-order processing: the problem of artificial intelligence	416
12.9 Concluding remarks	418
References	421
Index	467

PREFACE

This text is the result of two complementary experiences which I had in 1987 and 1988. The first was the opportunity, which I owe to Claude Godrèche, of delivering, in a pleasant seaside resort in Brittany, a series of lectures on the theory of neural networks. Claude encouraged me to write the proceedings in the form of a pedagogical book, a text which could be useful to the many people who are interested in the field. The second was a one-year sabbatical which I spent at the Hebrew University of Jerusalem on a research program on spin glasses and neural networks. The program was initiated by the Institute for Advanced Studies and organized by a team of distinguished physicists and biologists, namely Moshe Abeles, Hanoch Gutfreund, Haim Sompolinsky and Daniel Amit. Throughout the year, the Institute welcomed a number of researchers who shed different lights on a multi-faceted subject. The result is this introduction to the modeling of neural networks.

First of all, it is *an introduction*. Indeed the field evolves so fast that it is already impossible to have its various aspects encompassed within a single account.

Also it is *an introduction*, that is a peculiar perspective which rests on the fundamental hypothesis that the information processed by the nervous systems is encoded in the individual neuronal activities. This is the most widely admitted point of view. However, other assumptions have been suggested. For example M. Abeles puts forward the hypothesis that information is actually embedded in correlations between neuronal activities. How these two approaches are complementary or are exclusive of one another remains to be decided. The word '*modeling*' appears in the title. It must be understood in the sense which physicists give to it, that of a theoretical construction using as few ingredients as possible, aiming at accounting for a large number of properties and, one hopes, endowed with predictive power. Modeling, however, can be given another meaning, that of the rebuilding of a natural system as faithfully as possible. This is the sort of modeling that many biologists have in mind. And this is a perfectly legitimate approach. Biologists such as Kandel or Alkon strive to decipher the nature of neuronal signals by studying 'simple' molluscs, and their contributions to the field have been very important. In particular this strategy is convenient for the clarification of the basic mechanisms of neural dynamics such as electrical signal processing or synaptic plasticity. It is tempting to apply

the knowledge so acquired to the modeling of organisms simple enough for their neural apparatus to be fully documented. There exist nematodes whose neural system comprises only a few tens of neurons. Their connectivity graphs and the nature of synaptic connections are known. A roundworm, *Caenorhabditis elegans*, for example, has exactly 302 neurons connected by about 5000 chemical synapses and 2000 gap junctions. This knowledge is not enough, however, for toy models to show even the simplest behaviours of the animals (although encouraging results have been obtained by Neibur and Erdős as regards their locomotion). In reality physicists face the same type of problem: there exist accurate theories of systems comprising a few particles, but accuracy dwindles as the number of particles increases. However reliable predictions on average properties can be made when the system is homogeneous and the number of particles becomes so large that statistical tools may be used. This improvement is a manifestation of the central limit theorem. Considering the very large number of neurons that make the central nervous system of mammals for example, a statistical approach of a theory of neural networks is very appealing. The properties of the networks may be viewed as collective properties and individuals (particular neurons) do not matter. This is a point of view physicists are familiar with and this is the main guideline that is used in the forthcoming developments.

Finally the text deals with '*neural networks*'. What sort of neurons and what sort of networks are at stake here?

Theoreticians, when they are addressing a biological audience, are very careful to make it clear that they work on 'formal neurons', entities so remote from real neurons that there is no danger that results could be contradicted by observation. These neurons being so formal, they ought almost to be called probabilistic threshold automata and the theory considered simply as a branch of applied mathematics! But this is not fair. Formal neurons are caricatures, but they are caricatures of real neurons and the final if distant goal is the understanding of real, complex nervous systems. The general philosophy of the physicist's approach is to reveal all the implications brought about by a simple model and to make it more complicated only under the pressure of experimental observation. This attitude is adopted to model neuron used in this text.

It is also adopted for the types of network architectures one starts with. Most of the networks which are considered in this book are very simple homogeneous structures, either fully connected networks or feed-forward networks. It is certainly of paramount importance to take the true structure of real neural systems into account in order to explain their properties. To give some insight into how complex nervous systems may be organized, the first part of the text is devoted to a rapid overview of the anatomy of the central nervous system of man. But it is hopeless

to start a theory of neural networks with such complicated structures. It seems more fruitful to work first with rather amorphous networks, to study their properties and to see how these properties are modified by the introduction of some simple organizing principles.

Let us now devote a few words to the general spirit of the text.

The material it contains addresses two groups of putative reader. The first consists of physicists and applied mathematicians who are interested in the challenge of understanding the functioning of neural systems. For these people it is necessary to have some information as regards the biological implications of neuronal modelings. Chapter 2 has been written with this in mind. The second group is the growing population of neurobiologists who want to go beyond the traditional approach of their discipline, which is mainly descriptive in nature. For those people this book is certainly not easy reading, but they must accept that the mathematical developments are not given for the pleasure of doing mathematics! The text is so written that it can be understood by a graduate student in biology endowed with a little knowledge of algebra and a fair amount of courage. But what is to be gained is certainly worth the effort. To make things easier the book is organized along two levels.

Level I contains information which is most relevant to the theory of neural networks. It is written with standard fonts. The concatenation of all these parts makes a complete text in itself. Digressions not directly following the main stream of the text or mathematical developments are treated at level II. These parts are printed with smaller fonts. They could have been gathered at the ends of the corresponding chapters, but I found it more convenient to leave them at the places where they are most useful. However, if one wants to keep a feeling of continuity through the text it is better to skip them at a first reading. All the necessary technical material can be found in those parts, so that the book is self-contained. Moreover, I never hesitate to detail calculations. In my opinion the time spent in rebuilding chains of logical reasonings from sparse information is time lost.

The book presents the status of the art in the field of neural networks that has been reached by the end of 1988. Many authors have contributed to these early stages the theory, each with his own culture and personality. This diversity, which is hopefully reflected in the book, makes the subject most attractive. The field deals with several major disciplines: neurosciences obviously, but also experimental psychology, statistical physics and applied mathematics. The recent advances of the theory have raised hopes that artificial intelligence and experimental psychology or cognitive sciences and neurophysiology, which are still felt today as compartmentalized disciplines, may be soon understood in a

unique framework. To make progress in that direction, the many ideas scattered throughout the literature are gathered in the text. A selection from the available results has had to be made, however: the emphasis of the book is more on concepts than on the efficiency of related algorithms. The reasons for this are first that absolute efficiency is hard to measure and second that a concept can be the seed of unexpected advances even though its applications to specific cases have been disappointing. For example, rather unexpectedly, it happens that neural networks can solve combinatorial optimization problems. In particular, Hopfield and Tank have put forward a means of using neural networks to solve the traveling salesman problem. Unfortunately, the performance of their algorithm compares unfavorably with those of classical algorithms, but Durbin and Willshaw have devised another type of neuronal algorithm called the elastic algorithm which performs as well as the classical algorithms. Their idea is related to a concept of neuronal self-organization proposed by Kohonen in quite a different context. This example shows how important it is to give an account of as large a range of ideas as possible rather than to focus on a peculiar technique.

The overall aim of this book is to give enough background for the interested researcher to find his way through the proliferant literature and to enable him to bring his own original contribution to the field.

ACKNOWLEDGMENTS

These pages are the result of three favorable environmental factors, the summer school on statistical physics at Beg-Rohu in Brittany, the Institute for Advanced Studies in Israel and the Commissariat à l'Énergie Atomique in Grenoble. I thank their respective scientific managers, C. Godrèche, M. Yaari and J. Chappert, who gave me the opportunity and the means to work in this new field.

The rare privilege granted by this subject is that it enables one to meet interesting people from so different disciplines. It is obviously not possible to cite all of them. I simply give here a few names of scientists I actually met and who influenced me in the way I have considered the subject. One of them was the late P. Delattre who, about fifteen years ago, dared to propose that theoretical biology, a piece of nonsense in France at that time, could be considered as a genuine discipline. I have also been in touch with some of the very pioneers in the theory of neural networks such as J. Hérault, C. Von der Malsburg and E. Bienenstock. Then came the famous book by J.P. Changeux, *L'homme neuronal* which gave me both a source of inspiration and the certainty that neural networks were a fascinating field to explore. Meanwhile I discovered a little of the world of experimental psychology thanks to P. Delacour and I became convinced that many observations of experimental psychology could be amenable to analysis in the framework of neural networks theory. I owe much to the group of the École Normale in Paris. There, two people rapidly showed an interest in neural networks, namely G. Weisbuch and G. Toulouse. They did much to make the subject a legitimate field of research. They have been joined since then by M. Mézard and J.P. Nadal. The creativity of the whole group, together with the contribution of B. Derrida of Saclay, has brought much life to the subject. More recently I was also much impressed by the contacts I had with the Israeli groups. These acute people probably brought the most important contributions to the theory of neural networks, after those contained in the seminal papers of J.J. Hopfield. I particularly thank D. Amit, to whom I owe the opportunity to spend a year in Jerusalem; H. Gutfreund who combines rare scientific abilities with warm personal relationships, and E. Domany of the Weizmann Institute. Many thanks also go to my friend J.J. Niez, to M. Gordon, to A. Hervé, to I. Yekutieli and to all these people I worked with during the past few years. I am especially grateful to R. Van Zurk

and his electronics collaborators A. Mougins and C. Gamrat for the good time they gave me building early prototypes of neurocomputers.

Finally this book would have not come to its end without the almost unlimited patience of A. Orelle and C. Argoud-Puy, who typed the text, and that of Y. Gamrat, who corrected my English.

INTRODUCTION

1.1 Mind as an emergent property of nervous systems

1.1.1 Three positivist approaches to mind

Mind has always been a mystery and it is fair to say that it is still one. Religions settle this irritating question by assuming that mind is non-material: it is just linked during the duration of a life to the body, a link that death breaks. It must be realized that this metaphysical attitude pervaded even the theorization of natural phenomena: to ‘explain’ why a stone falls and a balloon filled with hot air tends to rise, Aristotle, in the fourth century BC, assumed that stones house a principle (a sort of a mind) which makes them fall and that balloons embed the opposite principle which makes them rise. Similarly Kepler, at the turn of the seventeenth century, thought that the planets were maintained on their elliptical tracks by some immaterial spirits. To cite a last example, chemists were convinced for quite a while that organic molecules could never be synthetized, since their synthesis required the action of a vital principle. Archimedes, about a century after Aristotle, Newton, a century after Kepler, and Wöhler, who carried out the first synthesis of urea by using only mineral materials, disproved these prejudices and, at least for positivists, there is no reason why mind should be kept outside the realm of experimental observation and logical reasoning.

We find in Descartes the first modern approach of mind. This author correctly placed the locus of thought in the brain but incorrectly restricted its domain to the internal (limbic) parts of the brain. He put forward the theory that information is trapped by sensory organs and conveyed to the central unit through nerves. Some decision is taken by the unit and transmitted to the effectors. This picture has never been dismissed, but a question arises: Who manages the central unit? Assuming that the task is carried out by the so-called homunculus no doubt only puts the problem of the nature of mind one step further on.

At present we have no answer regarding the nature of mind, but an impressive amount of knowledge has been accumulated over the last hundred years or so. The result of this quest has been the emergence of three major disciplines, neurobiology, (experimental) psychology and artificial intelligence (AI). These disciplines have grown apart, and, until

recently, there have been very few links between them, but results encompassing two disciplines, mainly biology and psychology, are now on the increase, raising hopes that a unifying framework may be devised in not too remote a future.

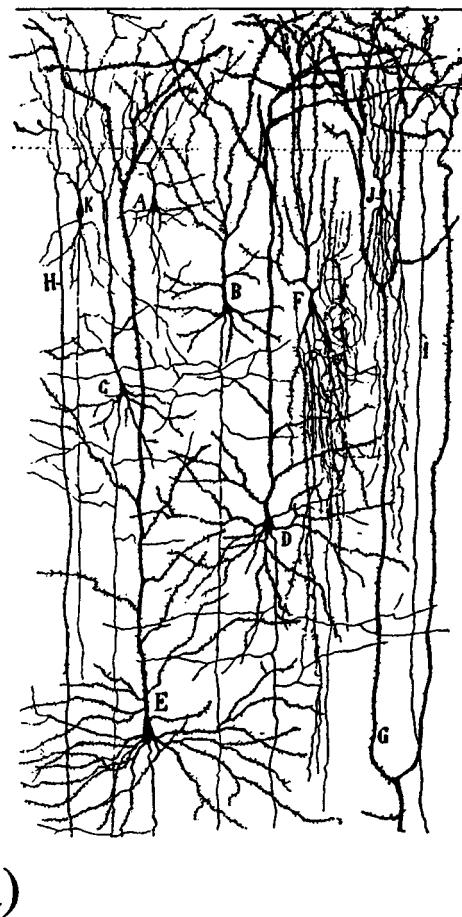
1.1.2 Neurobiology

Once it was recognized that the nervous networks are the systems which carry and process information in most forms of animal life, the next step was to determine the biophysical principles of their functioning. The first major discovery was made by Galvani, who in 1791 recognized the electrical nature of the nervous signals. Then came the fundamental contribution of Ramon y Cajal at the turn of this century, who, using a staining technique accidentally discovered by Golgi, showed that the neural system is made of an assembly of well-defined cells which he called neurons (see Fig. 1.1.a). The neurons communicate through tiny processes, the synapses (Fig. 1.1.b).

Five decades later the mechanisms involved in the creation and the propagation of neuronal electric signals were explained by Hodgkin and Huxley. Synaptic transmission was thoroughly studied, in particular by Katz. Today, much is known about the various molecules, the neurotransmitters in particular, and the complex proteins, such as the neuroreceptors, intervening in neuronal processes. Also, the reactions in which these molecules are involved, together with the ionic processes which are at work in neuronal dynamics, have been the focus of active research and knowledge is expanding fast in these areas.

1.1.3 Experimental psychology

Experimental psychology started in 1906, with the work of Pavlov on classical conditioning. His approach prompted a school of thought promoted in the thirties by Skinner. Behaviorism, as it is called, considers the neural system as a black box in order to avoid the experiments being biased by prejudices regarding the functioning of the internal structure of the box. The achievements of experimental psychology have since been impressive, and, in spite of a loss in popularity by behaviorism, its results and methods are still valid and widely used. It is obviously impossible to account here for the numerous developments in the discipline. Let us mention only the recent introduction of precise delay measurements. Nowadays delays between stimuli and responses are measured with an accuracy of the order of 1 ms. This technique gives insight into the architecture of the network and provides quantitative data that models have to account for.



a)

Figure 1.1.a. Structure of the cortical tissue.
Sagittal cut of the cortical sheet (After Ramon y Cajal).

1.1.4 Artificial intelligence

Fast computers have been available for four decades. Their versatility encouraged scientists to envisage the possibility that computers could mimic human behavior even in very elaborate contexts such as when playing chess. The corresponding discipline, called artificial intelligence, was first developed at MIT in the sixties, in particular by Minsky. A fair amount of success has been obtained in various domains such as pattern or speech recognition, problem solving or game playing. The best achievement of AI is expert systems. This is a hybrid approach, however, since the expertise is human and this success can therefore be considered

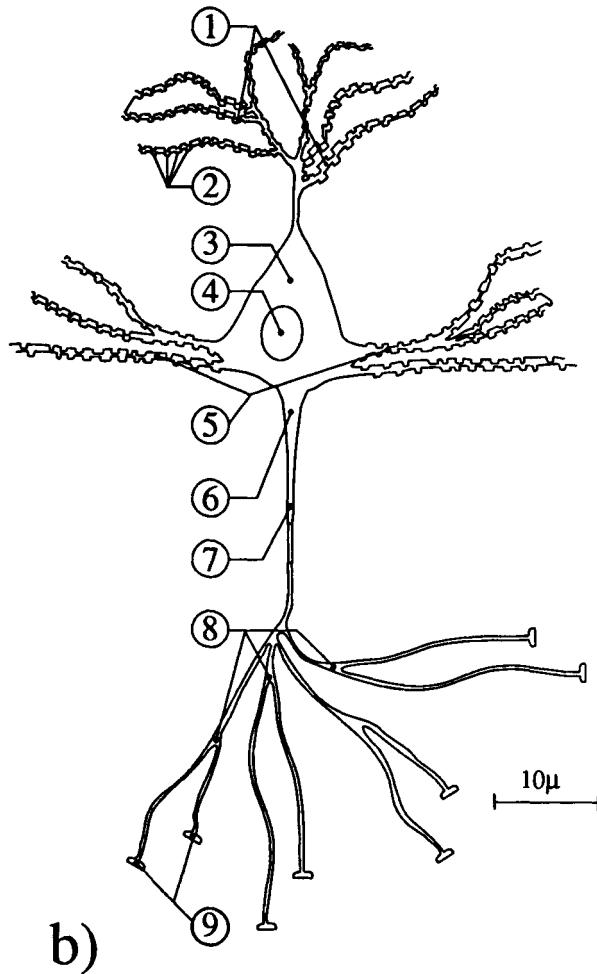


Figure 1.1.b. Structure of the cortical tissue.

Anatomy of a pyramidal neuron.

- 1: Apical and 5: basal dendrites. 2: Synaptic spines.
- 3: Soma. 4: Nucleus. 6: Hillock zone. 7: Axon.
- 8: Axonal tree. 9: Synaptic buttons.

as a sign that something essential is missing in the way AI tackles the problem of intelligence. Up to now solutions of AI problems have necessarily used programs written for serial machines. This approach has been criticized by Von Neumann who, even before AI had been invented, pointed out in *The Mind and the Brain*, a book published in 1957, that

the serial dynamics of computers and the corresponding architectures are far too remote from those of real neural networks, and that mimicking the behavior of humans (or of those animals who are endowed with complex neural systems) would probably require different designs.

1.1.5 Mixed approaches

Let us now consider briefly the advances achieved by using cross-disciplinary techniques.

The study of neurophysiological responses to external, well-defined situations, a technique that links neurobiology to experimental psychology, provided most interesting results. The observations made by Hubel and Wiesel in 1962, of selective responses in the striate cortex of cats and monkeys to simple geometrical patterns, and those carried out earlier, in 1957, by Mountcastle in the somato-sensory cortex of cats, led to the notion of a columnar organization of the cortical sheet. On the other hand, Kandel showed at the beginning of the seventies, on *Aplysia*, a marine mollusc, that the neurological parameters are modified by conditioning. This observation backed up the suggestion put forward in 1949 by Hebb, a psychologist, that conditioning could modify the synaptic efficacies in a definite way, thus providing a physiological basis for learning. According to Hebb the efficacy of a given synapse increases when the neurons the synapse links are simultaneously active.

It must be said that the relations between AI and psychology have been far less fruitful: the various mechanisms, for example in pattern processing, suggested by experimental psychology have not been taken into account in the building of AI programs, at least not up to now. AI, or more precisely one of its related disciplines, cybernetics, pervades the theorization of psychology in the form of block diagrams, an approach with weak predictive power.

Finally, there remains the ambitious goal of explaining the complicated behavior of natural neural systems, as observed by experimental psychology, for example using the elementary processes provided by neurobiology. The field is immense and is practically a complete *terra incognita*. Attempts made by Marr in vision processing showed that, even for this specialized task, the problems were much more difficult than was first imagined. There has however been some progress in that direction. Advances have mainly concerned the modeling of one of the major issues of psychology, for example, memory, namely the way it works and the way memorization proceeds. Most issues which will be developed in the following sections are concerned with this restricted albeit important neuronal function.

1.2 Neuronal nets as automata networks: a brief historical overview

1.2.1 Neurons as logical gates

The first modeling of neurons was devised by Pitts and McCulloch in 1943. In their view a neuron, which they called a formal neuron, is a logical gate with two possible internal states, active and silent. A neuron has a few entries provided by the outputs of other neurons. The entries are summed up and the state of the neuron is determined by the value of the resulting signal with respect to a certain threshold: if the signal is larger than the threshold the neuron is active; otherwise it is silent (Fig. 1.2).

It was shown that all logical (Boolean) operations can be implemented using conveniently associated formal neurons and well-chosen threshold values (see below). Therefore one could imagine that computers could be built by using neurons as basic components instead of transistors, thus making ‘neurocomputers’ universal machines.

These ideas were put forward at a time when the first electronic computers were being built. In reality even at this time nobody believed that brains worked like computers. For example, Von Neumann, whom we have already mentioned, has argued that Pitts and McCulloch networks, unlike real neural networks, are not robust. He suggested that the job carried out by one neuron could be shared by a set of neurons belonging to a pool. Robustness arises from redundancy.

More on formal neurons

The formal neuron of Pitts and McCulloch is a two-state automaton: the neuron j is either firing, $S_j = 1$, or it is silent, $S_j = 0$. The state S_j of neuron j is forwarded to every neuron i it is linked to. The influence of j on i is $J_{ij}S_j$, with

$$J_{ij} = \begin{cases} +1 & \text{if the interaction between } j \text{ and } i \text{ is excitatory,} \\ -1 & \text{if it is inhibitory.} \end{cases}$$

All influences impinging on i are added and the resulting ‘field’ h_i ,

$$h_i = \sum_j J_{ij}S_j, \quad (1.1)$$

is compared with a threshold θ_i :

$$S_i = \begin{cases} 1 & \text{if } h_i > \theta_i, \\ 0 & \text{if } h_i < \theta_i. \end{cases}$$

The main features of this model are still valid today, but during the period which saw the birth of electronic computers the emphasis was on logics and digital processing. It seemed important to prove that neurons could be used as elementary components in the building of logical gates in the very same way that transistors are. Pitts and McCulloch showed that this is indeed the case (see Fig. 1.3). For example:

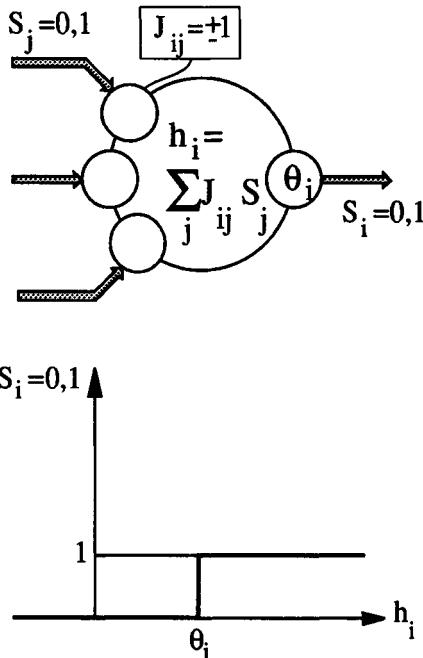


Figure 1.2. The formal neuron of Pitts and McCulloch.

- The NOT operation is carried out by a one-input neuron with

$$J_{ij} = -1, \quad \text{and} \quad \theta_i = -\frac{1}{2}.$$

- The AND operation is carried out by a two-input neuron with

$$J_{ij} = J_{ik} = +1 \quad \text{and} \quad \theta_i = \frac{3}{2}.$$

- The OR operation is carried out by a two-input neuron with

$$J_{ij} = J_{ik} = +1 \quad \text{and} \quad \theta_i = \frac{1}{2}.$$

One knows that these three gates are enough to generate all the other logical gates (see Fig. 1.3 for the realization of an XOR gate). McCulloch concluded that any universal (Turing) machine can be assembled by using formal neurons as fundamental components. This is indeed an interesting result, but we have already stressed that a Boolean architecture is both fragile and precise, whereas the biological systems are robust and ill-adapted to chained logical reasonings. For example, one is easily lost by a string of three successive negations in a single sentence. The formal neurons were soon felt to be completely disconnected from the biological reality and dismissed on that account. The computational power of threshold automata networks was ignored for a while afterwards. History repeated itself some fifteen years later with the perceptron!

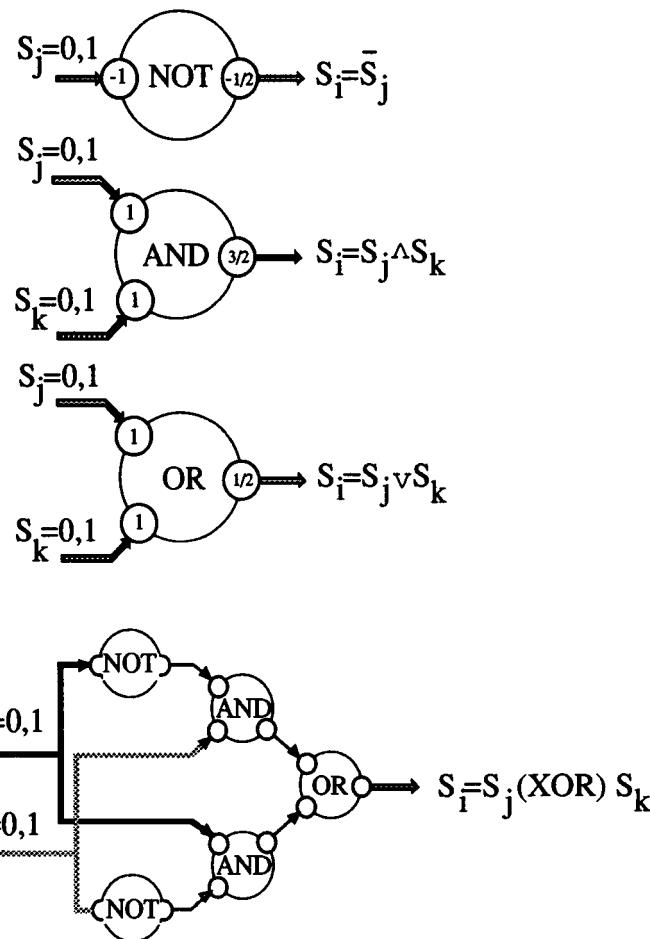


Figure 1.3. Formal neurons as Boolean gates and how to combine gates to build an XOR function.

1.2.2 Neuronal activities as collective processes

In 1954 Cragg and Temperley adopted the concept of neuronal pools of Von Neumann but they gave the idea a more physical meaning. They considered that the activity of a neuronal network results in collective behavior (similar to the onset of ferromagnetism, for example). This is a well-founded idea which is at the root of later developments in the theory of memory in neural networks.

In 1974 Little took a further step in that direction by introducing noise, thus making neural networks probabilistic automata networks. This was a welcome improvement because neurons are very noisy, unre-

liable devices. It was therefore legitimate to wonder how such inaccurate systems could manage to deliver definite signals. Collective behavior was the answer, because noise does not severely damage the average properties of a collective system as long as it does not exceed a certain critical value.

It was Hopfield in 1982 who was the first to fully recognize that memory is a collective process, whose meaning is the result of the interplay between a large number of elementary units, namely the neurons, and that memorization could be achieved by giving to synaptic efficacies the strengths which follow the simple prescription advocated by Hebb. The prescription is called the Hebbian rule. (More precisely, Hopfield used a symmetrized version of the Hebbian rule. See section 4.2.) Instead of settling in one or two global states as in ferromagnetism, the system, as a whole, has several stable states (two for each memorized pattern). Which one is actually retrieved depends on the state from which the system starts. The model is reminiscent of spin glasses, a much-studied model of magnetism with random interactions. Although synaptic plasticity in cortical areas still lacks firm experimental observations, the Hopfield model can be considered as being the first mathematically well-defined model aiming at bridging the gap between physiology and experimental psychology.

1.2.3 Learning

As already stressed, the central idea behind all these models is that the neural activity is a collective phenomenon involving a strong connectivity between all the elementary units comprising the system. In its extreme form, in particular that adopted by Hopfield, all neurons are interconnected. The interplay between the units of such a recurrent network makes it possible for the neuronal states to change, possibly to settle in meaningful configurations, even in the absence of input activities. But there exists another tradition, closely related to behaviorism, which only considers feedforward interactions. Feedforward neural nets do not display any oriented loop of interactions. In those systems there is a flow of information which runs from input units, that is to say from neurons which are not fed by any other neuron of the net, to output units, that is to say to neurons which do not feed any other neuron. The downstream states of the net are fully determined by the activities of input neurons.

Learning is a process which settles the parameters of the neural system, the synaptic strengths J_{ij} (whose number is of the order of N^2) and the N thresholds θ_i , so that the network yields wanted outputs to given inputs. The problem of learning is much more difficult to solve in recurrent networks than it is in feedforward networks. This is the reason why learning has been studied mainly in the latter type of network.

The first learning algorithm was proposed independently by Rosenblatt in 1958 on the one hand and, in a different context, by Widrow in 1961 on the other. It is based on the paradigm of reward and punishment and is called the *perceptron* algorithm. The perceptron is the name of the machine in which Rosenblatt implemented its learning rule. The name Widrow gave to his machine was *adaline*. Perceptron and adaline are two-layered, feedforward networks. These systems were intended to carry out categorization tasks and it was proven that if sets of parameters exist which enable the machine to do the task correctly, the perceptron algorithm finds one of the sets. However, Minsky and Papert showed in 1968 that the sets of items which a perceptron is able to categorize are severely limited by the inability of the perceptron to perform certain logical operations such as the XOR. After this setback, research on the perceptron was almost completely abandoned.

Giving up the idea of perceptron, Kohonen, in 1972, suggested returning to the basic Hebbian associative mechanism to implement a learning rule in bilayered neural systems. In actual fact similar concepts can be found in an earlier work of Willshaw, Buneman and Longuet-Higgins published in 1969, but it was Kohonen who first emphasized the possible applications of the idea of associative memory. He also improved the performances of his design by appealing to the projection algorithm, a mathematical trick due to Penrose.

A natural extension for feedforward systems was to add extra layers called hidden layers to the two layers, the input layer and the output layer, considered so far. This gave rise to several developments. One, promoted by Fukushima, was the neocognitron, an enlarged version of the perceptron. A further development was the introduction of continuous variables which enabled the use of derivatives in the process of error minimization. The back-propagation algorithm, which is derived using this optimization process was put forward simultaneously, at the beginning of the eighties, by several authors, Rumelhart, Le Cun and Parker. Finally, Hinton and Sejnowski suggested that the hidden layers could be used to embed an internal representation of the patterns experienced by the visible layers. This attractive idea was implemented in their Boltzmann machine. Unfortunately this algorithm proved to be too slow.

Meanwhile another approach to the problem of learning, that of self-organization, was explored. It started with the pioneering work of C. Von der Malsburg in 1973 on a model of feature detectors. In 1982 Kohonen devised an algorithm which realizes mappings between spaces of different dimensionalities and more recently, in 1986, Linsker published a model which explains how a whole series of visual feature detectors could emerge from simple learning rules applied to topologically

constrained neural networks.

All learning algorithms, with the exception of the Boltzmann machine we have quoted, deal with feedforward connections. Natural neural networks are neither feedforward systems nor fully connected systems. One could say that the sensory tracts or the effector pathways are more similar to directed networks and that central associative cortical areas are more intertwined systems. Nevertheless, there is a need to devise learning algorithms in strongly connected networks. A step in this direction has been made by Wallace and Gardner, who suggested an extension of the perceptron algorithm to fully connected networks. These different topics will be discussed in the text.

1.3 Organization of the book

This book is divided in four parts:

- *The first part is biologically oriented.* — It is intended for those who are quite unacquainted with neurobiological mechanisms and it consists simply of Chapter 2. It begins with an anatomical description of central nervous systems, which shows how well organized these wonderful constructions are. Then comes a brief introduction to neurophysiology. The basic phenomena thus described serve as building bricks for the modeling of neuronal dynamics as well as emphasizing the limitations of models. Finally, some observations on experimental psychology are presented which are used as guides in the search for a theory for learning.
- *The second part is devoted to the dynamics of neuronal states.* — It comprises Chapters 3, 4 and 5. The thesis advanced here is that the dynamics of neural networks is basically a stochastic (random) dynamics. This makes the systems amenable to classical treatments of statistical physics. The analysis is applied to the study of a simple type of model, the Hebbian, which accounts for associative memory. Associative memory is a crucial property of central nervous systems. (When we are asked to say how much ‘two times three’ is there is no multiplier in our brain to give the answer. Instead we associate the pattern ‘is six’ to the pattern ‘two times three’ in the same way as we associate the various parts of a song when we recall it.) The analysis of models of associative memory has been pushed further still. The main results are given here.
- *The third part deals with learning.* — This part encompasses Chapters 6 to 9. As already stressed, a neural network learns when the parameters which define the net are modified. These modifications may be controlled either by the activity of the system itself or by the influence of external actors. There exist several learning paradigms which are discussed in this part of the book. For example, we have already mentioned that learning can be the result of an association process, an

idea stemming from classical conditioning experiments. It can also be an error correction process, a concept favoured by cyberneticians. It can proceed by reward and punishment. Finally, it may be viewed as an optimization process which adapts at best the answers of the system to its environment.

- *The possible applications of neural networks are reviewed in the fourth part of this book.* — This part is comprised of Chapters 10 and 11. The theory of neural nets can first be considered as a tool for modeling biological systems. It is encouraging to note that this theory has been successfully applied in several instances to account for electrophysiological observations. Also, up to now, there have been no indications that the theory has led to untenable conclusions. It is obvious that the main emphasis would have to be on applications in artificial intelligence. It must be said that, for the moment, no real breakthrough has been made in that direction. Pattern recognition seems a domain well suited for neural networks, but we are still far from understanding such basic properties as invariant recognition, for example. (Invariant recognition is the property of a system to correctly label an object whatever its position, its distance from the detectors, etc.) Some interesting ideas, which are presented here, have however been proposed. Applications to expert systems by using layered neural networks have received more attention. Real success has been obtained, but the problem is that nobody yet knows why. If in some instances the system works satisfactorily, in others it fails for no apparent reason. The theory of learning and generalization is now the focus of active research. It is all the more probable that significant results will be obtained very soon. The neural networks have also been found to be an interesting tool to deal with combinatorial optimization problems. This is a vast domain, which could be an important outlet for these systems. By using the thermal annealing procedure, the neural networks are quickly able to find a good solution to a complicated problem, though it is generally not the best one. They are well adapted to give sensible answers to ill-defined problems. These properties make them good candidates for tackling many problems concerning real life, as, indeed they should be. The last section is devoted to possible architectures of neuronal machines called neurocomputers. The building of machines is under way in several countries. The availability of a powerful neurocomputer, able to run a hundred times faster than the fastest serial machine, will probably speed up the pace of research in the field of neural networks.

THE BIOLOGY OF NEURAL NETWORKS: A FEW FEATURES FOR THE SAKE OF NON-BIOLOGISTS

This chapter is *not* a course on neurobiology. As stated in the title, it is intended to gather a few facts relevant to neural modeling, for the benefit of those not acquainted with biology. The material which is displayed has been selected on the following accounts. First of all, it is made of neurobiological data that form the basic bricks of the model. Then it comprises a number of observations which have been subjects for theoretical investigations. Finally, it strives to settle the limits of the current status of research in this field by giving an insight on the huge complexity of central nervous systems.

2.1 Three approaches to the study of the functioning of central nervous systems

Let us assume that we have a very complicated machine of unknown origin and that our goal is to understand its functioning. Probably the first thing we do is to observe its structure. In general this analysis reveals a hierarchical organization comprising a number of levels of decreasing complexity: units belonging to a given rank are made of simpler units of lower rank and so on, till we arrive at the last level of the hierarchy, which is a collection of indivisible parts.

The next step is to bring to light what the units are made for, how their presence manifests itself in the machine and how their absence damages its properties. This study is first carried out on pieces of the lowest order, because the functions of these components are bound to be simpler than those of items of higher rank. Assuming that this task has been carried out successfully, the following step is naturally to understand the function of more complicated units by using the knowledge we gathered on their components. Thus we hope to reach the top of the hierarchical organization via a step-by-step process.

The efficiency of this bottom-up strategy quickly deteriorates, however, as one goes away from the ground level of the hierarchy. The

reason is that the function of units belonging to a given level does not determine the function of the assemblies at higher levels (even though the functioning of the system as a whole is known). The building of a theory of larger systems makes it necessary that their functions are first made clear. This difficulty suggests a top-down strategy, which consists in observing the properties of the machine in its entirety and, in particular, its responses to given inputs, in the hope that these observations give some insights into its principles of functioning.

Central nervous systems are extremely complicated structures, but if we treat them as mere machines we can use the strategies we described above to strive to understand their functioning:

- The first approach aims at deciphering the organization of the system and, in so doing, defining its parts. This is the role of *anatomy*, which concerns itself with structure. As in man-made artifacts, anatomical observation of the central nervous system reveals a hierarchical organization. The difference with artificial constructions, however, is that it is not clear what the bottom level should be made of. Are molecules, cellular organites, neurons or groups of neurons the elementary entities? This is the point where the art and the limits of the modeler come in. The choice of the material we display in this chapter reflects the level of description we have adopted, which is to choose the neuron as the basic brick of the model. For example, we pay only little attention to neurobiochemistry and to molecular mechanisms involved in neuronal activity. These processes are obviously essential for the functioning of the system, but, it is thought, not for the functions themselves. Admittedly, we have been influenced by the fact that the basic components of computers are transistors and that, as far as the functional properties of transistors are concerned, the physical properties which make them work and the technology used for their making are of little importance.
- The second approach is to focus attention on the properties of the various entities which make the central nervous systems. This is the role of *neurophysiology*, which concerns itself with function. The neuronal cell is the main object of study, and research in this field has been so active that it has been said that the neuron is perhaps the best known of all cells. Neurophysiology obviously strives also to understand the functioning of higher-level neuronal assemblies, in particular in terms of the properties of neurons. But this is not an easy task and even the role of such a vital structure as the cerebellum is not elucidated.

This is why neurobiology invites the final approach, namely *experimental psychology*. Its object is to record the behavior of the entire organism, trying to make its responses to stimuli as reproducible as possible by using simple and well-controlled protocols and to induce some general organizing and functioning principles from these observations.

A short review of the three approaches to central nervous systems is given in this chapter. A very large body of knowledge is now available in neurobiology and the material which is displayed here is necessarily restricted. It concerns data which we feel is essential in the building of any realistic model of a neural network. We also insist on including features which we regard as relevant for the future development of the theory.

2.2 The anatomy of central nervous systems

2.2.1 *Why a section on anatomy?*

When observed through a microscope, the cortical tissue, stained by a convenient technique, displays a surprisingly amorphous and homogeneous texture. It is impossible for example to decide, from crude anatomical observation, from which part of the cortex a piece of cortical tissue has been taken.[†] It is even impossible, without using genetic investigation, to distinguish between samples coming from different species, be they rats, monkeys or man. It is this uniformity which is so appealing to physicists. Physicists indeed are used to work with systems made of large numbers of similar components, atoms or molecules. Physically different systems are essentially the same and display the same physical properties. Therefore it is legitimate to make quantitative predictions by carrying out averages (ensemble averages) over many identical systems. Neurons are so numerous, and look so similar with respect to each other, that it is tempting to apply the concepts and the methods of statistical physics to these biological tissues. It is this situation which prompted the recent interest in the study of statistical theories of neural networks.

On the other hand, the structure of central nervous systems, when viewed on a macroscopic scale, seems to be perfectly organized. Their architectures are obviously under the full control of the genetic code, which is different for different individuals. If one assumes that the central nervous systems are completely determined by the genetic code then the individuals are essentially dissimilar and averaging is meaningless. Each individual is to be studied *per se* and there seems to be no point in building theories of neural systems.

These two points of view are not incompatible, however. We already mentioned that the central nervous systems are hierarchically structured. There is a general agreement that the lower the level the less strongly its geometrical structure is genetically determined. Indeed the human brain comprises some 3×10^{10} neurons and 10^{14} synapses. Following Changeux,

[†] Thorough investigations reveal differences, however. The visual cortex, for example, is twice as thick as other regions of the cortex.

one can argue that complete genetic determinism would imply that the efficacy of every synapse is controlled by one gene and therefore that the number of genes devoted to the building of central nervous systems is of the order of the number of synapses. As pointed out by Changeux, this is completely unrealistic, at least in sufficiently complex animals such as mammals, and the consequence is that the genome is unable to control the lower levels of the hierarchical organization of the neuronal system. For example, the exact position of every ion channel in the neuronal membrane is surely irrelevant. To take this a step further, since so many neurons are continuously dying, the role of a single neuron is of little importance. This leads to the idea of Cragg and Temperley that the properties of neural networks are collective properties stemming from assemblies of neurons and that the presence or the absence of a given neuron does not make any difference.

However, the argument weakens as neuronal assemblies of higher levels are considered. Indeed, the number of degrees of freedom which determines the global architecture of these units decreases and the role of genetics increases. There is no doubt that a central nervous system does not perform correctly when it is deprived of its thalamus or of its reticular formation. Therefore there must exist a level in the hierarchy where genetics takes the lead. Where the turning point lies is not known. For example, the existence of a given cortical microcolumn seems not to matter, but the status of columns is less clear, although plasticity on a large scale has been observed. Along the same line of thoughts, one could suggest that evolution has not managed to make certain organisms cross the turning point, which could explain that the nervous systems of those creatures are fully wired and made of labeled neurons.

In our opinion these arguments show that it is legitimate to build a statistical theory of neural networks, as least as long as we consider these networks as small parts of large systems, say as cortical microcolumns. But they also show that it is necessary for the theory to take the real organization into account when larger systems are at stake. This is why, although this book is mainly concerned with amorphous or poorly structured systems, it starts with a description of the anatomy of the central nervous systems (CNS for short) of mammals.

2.2.2 *Level one: the general structure of CNS in mammals*

The highest level of the CNS of mammals shows a remarkable organization, which is not unlike the organization of large mainframe computers (Fig. 2.1):

- The central processor is the *cortex*, a flat structure which lies on the surface of the brain.
- The *thalamus* plays the role of a ‘frontal’ computer. All information

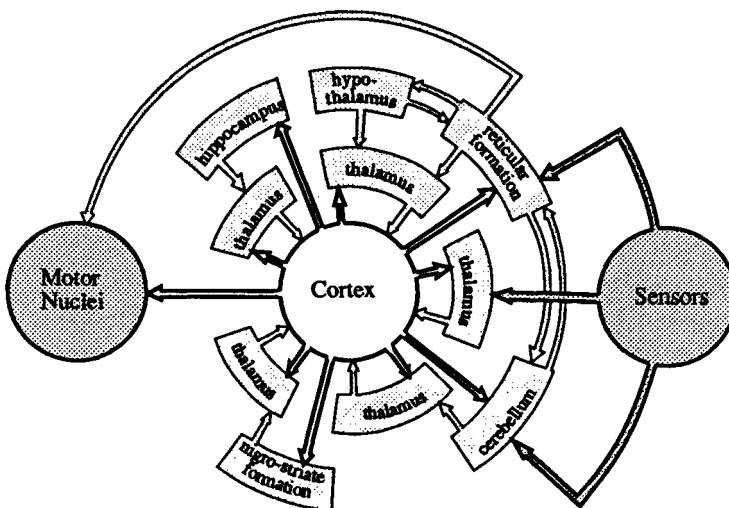


Figure 2.1. The architecture of the central nervous system (CNS) of mammals.

which flows to and from the cortex is processed by the thalamus. The thalamus is made of regions or nuclei, which make connections with specialized structures. For example, the signals coming from the eyes pass on their way to the cortex through a thalamic nucleus which is called the lateral geniculate nucleus (LGN).

- The *perithalamic structures* play ancillary roles, like slave computers. They are, no doubt, essential for the functioning of the CNS but it must be said that their roles in this complex machinery are not fully elucidated. The various perithalamic structures are the following:

- The *hypothalamus*. This structure embeds fundamental programs necessary for survival. It also controls hormonal secretions.
- The *reticular formation*. This coordinates and executes the hypothalamic programs.
- The *nigrostriate formation*. It is responsible for the coordination of long-lasting actions.
- The *cerebellum*. This is involved in the storing, the retrieval and the precise adjustment of sequences of coordinated motions.
- The *hippocampus*. Plays a key role in the storage process involved in long-term memory.
- The *colliculus*. It drives the ocular movements.

It is all the more probable that these structures carry out functions other

than those cited here. Figure 2.2 shows the organization of the CNS, its elements and the main links between the elements.

2.2.3 Level two: the cortical areas and the functional structure of the cortex

The cortex is a quasi-bidimensional structure made of two sheets, 2 mm thick, which, in man, each spread over about 11 dm^2 . The two sheets are connected through the corpus callosum, a bus comprising about 800 million fibers. The cortex is made of neuronal cells, the neurons which are the basic components of this architecture.

The outputs of neurons, at least those which lead out of the cortex, are conveyed along myelinated fibers and form the white matter. The grey matter is the cortex itself: as in computers, much space is needed for wiring.

The study of the functional consequences of traumas in man and those of the degeneracy of tissues following localized lesions in animals led to the notion of *cortical* (or *Brodmann*) *areas*.

A cortical area is defined both functionally and anatomically. Its destruction brings about the massive loss of a specific property of the system such as phonation. Thorough anatomical investigations, on the other hand, show that the structure of the neuronal tissue, i.e. the density of neurons, the width of the various layers of the cortical sheets, etc., varies from area to area. Also one area projects specifically onto another area. There are about 40 cortical areas in one hemisphere. The two hemispheres are not fully symmetrical, but the main organization such as the one depicted in Fig. 2.3 is essentially identical for both parts of the cortex.

The cortical areas are numbered using either figures (for example, the primary visual cortical area is area 17) or letters (V, for example, is used to indicate the visual cortical areas). This latter notation is more transparent and it is used here. Thus:

- V is used for visual areas;
- A for auditory areas;
- S for somesthetic areas;
- M for motor areas;
- T for temporal areas;
- F for frontal areas.

A1, M1, S1 and V1 are primary areas. A2, M2, S2 and V2 secondary areas and other associative areas are labeled by sets of letters. SA, for example, is an area which is thought to associate auditory and somesthetic signals. Figure 2.2 shows the various areas of the left hemisphere and the way they project onto one another. Figure 2.3, redrawn from a diagram of Burnod's, shows a reconstruction of the graph of connections. The graph appears to be almost planar and remarkably symmetrical. The cortex is organized around the somato-motor block. There are three concentric levels of connections:

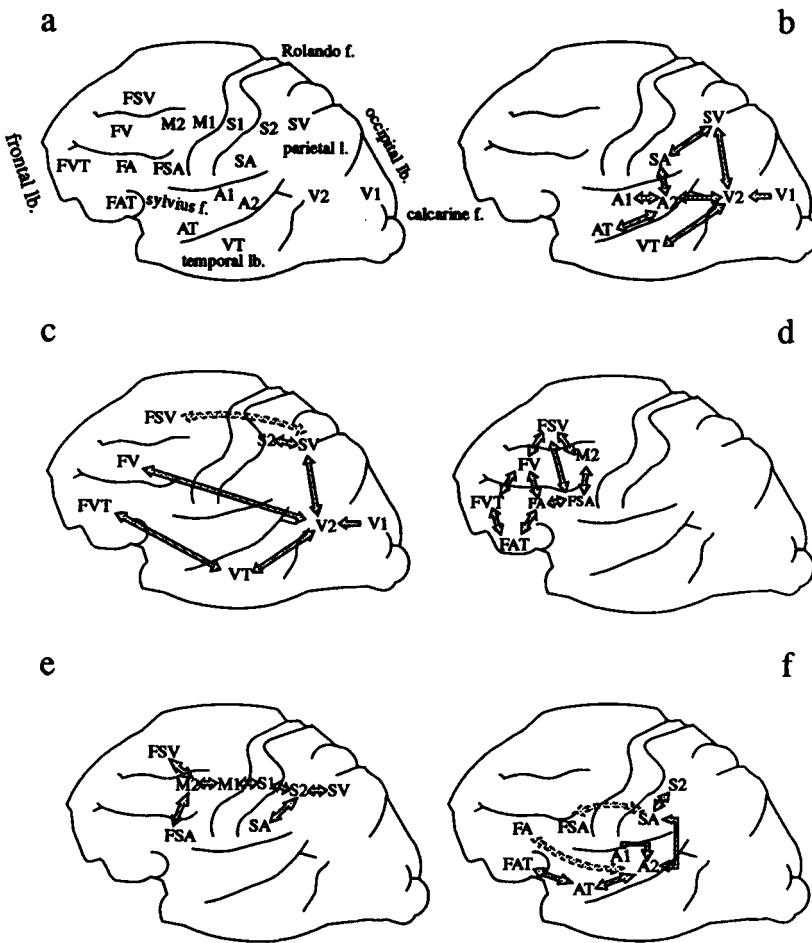


Figure 2.2. The cortical areas in the left hemisphere.
 a) Labeling of cortical areas and cortical pathways.
 b) and c) Between visual and auditory and visual and frontal.
 d) Motor and frontal.
 e) and f) Sensory-and-frontal, and auditory-and-frontal areas.

- the parietal level;
- the receptor level, and
- the temporal level.

Similarly, there are four radial systems of connection:

- the auditory recognition system;
- the language production system;

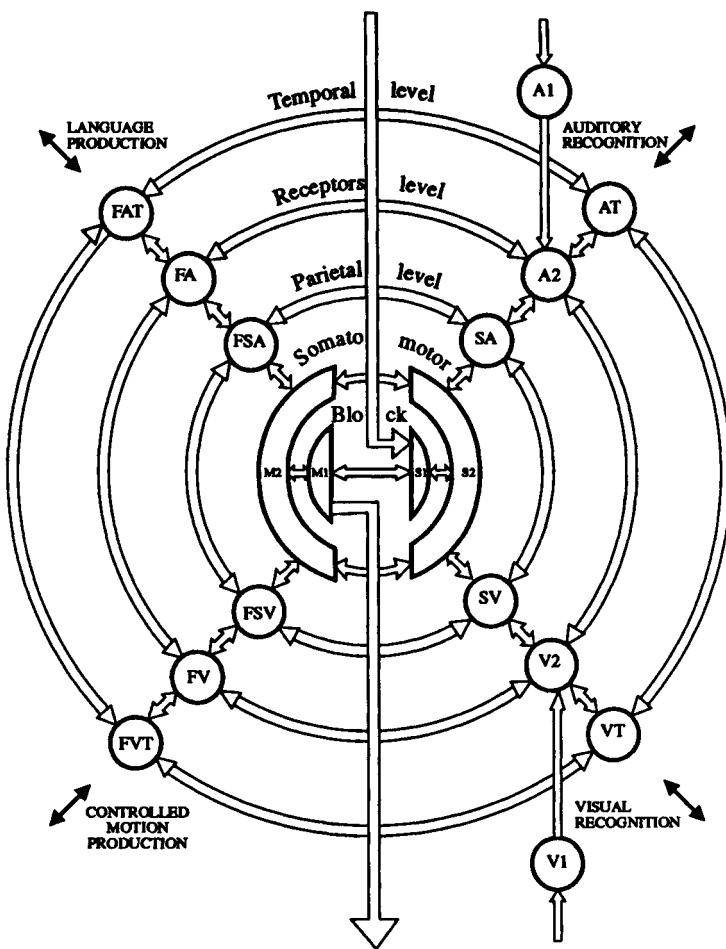


Figure 2.3. The functional organization of the left cortical hemisphere (Redrawn from Burnod).

- the visual recognition system, and
- the visually controlled motions production system.

Dotted lines on Fig. 2.2 indicate connections which observation failed to reveal. The graph is not complete. It must be connected to a simpler graph corresponding to limbic areas located in the inner surface of the hemisphere, that which lies in between the two parts of the brain. These limbic areas receive signals from the various nuclei of thalamus and from the olfactory bulb.

The graph of inter-area connections is determined genetically. Genes promote the making of specific tracer molecules which pave the way along which the nerves grow during epigenesis. The tracer molecules trigger the formation of a protein, the nerve growth factor (NGF), which controls the spreading of neuronal fibers (the neurites).

2.2.4 Level three: cortical maps and cortical bands

The primary areas receive signals from sensory organs. The topology of this information, the neighborhood of elements which form the sensory message, is mirrored in the cortex itself, thus providing *cortical maps*. For example, area A1 receives its information from the cochlea (through the specific thalamic nucleus). This organ, which lies in the internal ear, is a frequency analyzer. A1 is an auditory map with regions, the *cortical bands*, specifically sensitive to a certain frequency. Topology conservation manifests itself by bands organized along increasing frequencies, a mechanism which yields an internal picture, a map, which reflects a regularity of the outside world. This phenomenon, called tonotopy, is rather surprising when one considers the number of synaptic contacts the signal has to pass from the ear to the primary auditory cortex. Similar orderings are found in the primary visual cortex: nearby retinal receptors excite nearby cortical neurons. This is retinotopy. Somatotopy is also found in S1 (and M1), resulting in an internal representation of the body on the primary somatic cortical area. Models of topological maps are discussed in section 9.1.3 (the algorithm of Kohonen) and in section 9.2.1 (the early model of Von der Malsburg).

The cortical bands are a few millimeters long. Their widths range from 30 to about 500 microns. The bands are defined functionally: they respond specifically to given patterns of incoming activities or *features*. Characterizing the features is a difficult but essential task for the neurophysiologists. There are indications regarding the way features are biologically defined, mainly for primary areas where the task is the easiest. Color, motion direction and bar orientations have been clearly established as features which trigger activity in specific regions of the primary visual cortex. However, no process which would detect the presence of corners into images has been reported so far. The approach of Linsker to the natural emergence of feature detectors in the visual cortex is discussed in section 9.2.2.

The cortical maps are not simple geometrical transforms of sensory pictures, but rather deformed versions of environment. They emphasize important regions, giving, for example, more space to the representation of the center of the retina, the fovea, than to more eccentric regions. Also, fingers are given more space in S1 than arms. Neither are the cortical maps simple homeotopic transforms of sensory maps. A retinal image,

for example, is cut vertically in two halves, each of these being sent, through the optic chiasma, to a different brain hemisphere. Moreover, a sensory map gives rise in the secondary areas to several cortical maps, which are processed partly side by side, partly in series. In the secondary visual area V2 of squirrel monkeys at least ten maps have been observed. There are probably even more visual maps in the cortex of man. Each map is supposed to process a specific sort of feature or a well-specified combination of features of the retinal image. It seems that, in V2, geometrical features, color on the one hand and motion features on the other, are treated in parallel through parvo- and magno-cellular pathways. Multiple maps have also been found in S2.

The existence of maps is determined genetically, which means that there exists a mechanism which compels the cortex to be organized according to topology preserving principles. However, the precise geometry of the maps is not determined by the genes. The region of S1 devoted to a given finger of a chimpanzee disappears once the finger is amputated. This region is invaded by neighboring bands.

The general principles of band organization are the following:

- The orientation of the bands is perpendicular to that of the *cortical fissures* with which they are associated. There are three main fissures, the Rolando fissure which separates the fore cortex from the rear cortex, the Sylvius fissure which delimits the temporal lobe and the Calcarine fissure in the occipital cortex (see Fig. 2.2). There also exist secondary fissures and it is the structure of this set of fissures that drives the geometry of the graph of connections between the cortical areas.
- Bands tend to split along their large dimensions, giving rise to narrower bands. The splitting process stops when the bands are about 30 microns wide. It is assumed that splitting is a means for the cortex to store more information, since the process is triggered by learning.
- When two feature detectors oriented by the same fissure compete in a given map, they share the space by alternating the bands devoted to each feature. Ocular dominance is an example.
- Two features which occupy a same cortical territory can excite two different layers of the cortical sheet and therefore avoid competing. For example, the receptive field of a given retinal receptor is determined by those cells of layer IV in V1 which are excited by the receptor. They do not interfere with the bands sensitive to bar orientations, which comprise neurons of layers II and III. (For the definition of cortical layers see Fig. 2.5.) Figure 2.4 materializes the classical model of cortical organization by Hubel and Wiesel. Zeki has shown that in actual fact the organization is more complicated. There may exist several systems of features, say bar orientations and colors, competing for the same cortical

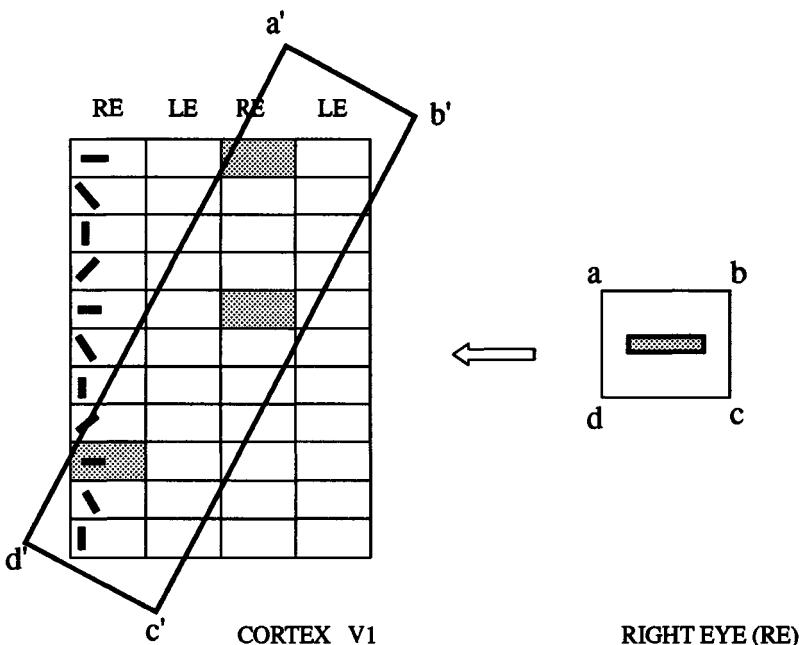


Figure 2.4. Cortical bands in area V1. Horizontal stripes are bands sensitive to line orientation. Vertical stripes are bands of ocular dominance. A retinal region $abcd$ in the right eye (RE) stimulated by a horizontal line projects on a receptive field $a'b'c'd'$ and gives rise to cortical activities in the shaded areas.

area. When the case arises the bands break into smaller pieces or ‘blobs’, which respond to those specific features. Thus Zeki has found in area V1 blobs sensitive to orientation intertwined with blobs sensitive to colors.

2.2.5 Columns and microcolumns

The concept of bands and that of *columns* and *microcolumns* are closely related. The former, as we have seen, is functional in nature, whereas the latter is anatomical. The main neuronal population in the cortex is that of *pyramidal neurons* (see Fig. 1.2). These neurons have two sorts of afferent fibers, the basal dendrites and the apical dendrites. These fibers collect signals and therefore their extension is of functional significance. The diameter of the nearly isotropic tree of basal dendrites is around 500 microns. The apical dendrites are oriented and their elongation stretches over a few millimeters. These extensions taken together determine a region called a column, which is therefore somewhat similar to a band of maximum extension. One of the important properties of columns is that the efferent fibers of the neurons of a column, the axons, make contact

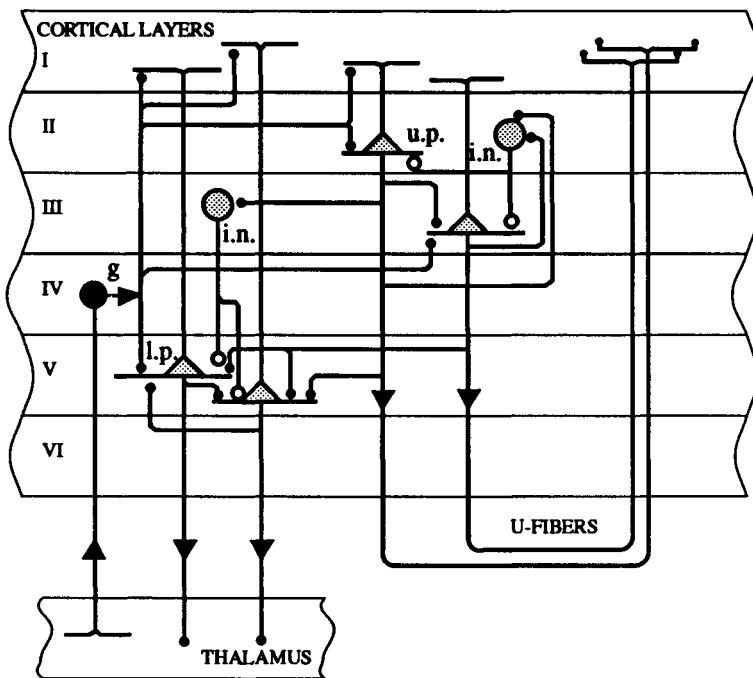


Figure 2.5. The neuronal structure of the cortical sheet.
 i.n.: inhibitory neurons; g: granule cells; l.p.: lower pyramidal; u.p.: upper pyramidal neurons; solid circles: excitatory synapses; open circles: inhibitory synapses.

only with other specific columns, some of them very far away, columns of another cortical area for example.

Microcolumns are structures which are better anatomically defined. They are sets of strongly connected neurons which develop during epigenesis. The fibers, dendrites and axons, of neurons belonging to a given microcolumn form the so-called fiber bundles. The microcolumns comprise about 100 fully connected neurons (some authors claim that the number is even lower, of the order of 30). The diameter of microcolumns is approximately 30 micrometers: this is the lower limit of the width of bands. Electrophysiological experiments have shown that the activities of the neurons of a microcolumn are strongly correlated: it is difficult to excite or inhibit a neuron without modifying the activities of all other neurons of the microcolumn. The basic functional unit of the cortex seems therefore to be the microcolumn. A band is a juxtaposition of

microcolumns up to a size corresponding to that of a column.

To summarize, the cortex comprises about:

3×10^{10} neurons,	3×10^5 columns,
3×10^8 microcolumns,	10^3 maps,
5×10^6 cortical bands,	50 cortical areas.

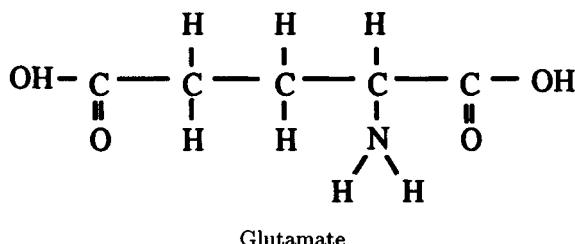
The neuronal density is 6×10^4 neurons mm^{-2} . (All figures within a factor of 2 or 3.)

2.2.6 Level four: the transverse structure of the cortical sheet

Histology shows that the cortex is made of six *layers*, labeled I to VI, the thickness of which varies from cortical area to cortical area. The structure of the layers is remarkably uniform throughout the cortex. Layers differ by the type and the density of the neurons they are made of. Anatomists tend to distinguish many sorts of cortical neurons, but the gross organization relies on only two types of neurons: the pyramidal neurons which account for more than 80 % of the neuronal population and the rest which is the set of interneurons.

There are two families of pyramidal neurons, the *upper pyramidal neurons* lying in layers II and III and the *lower pyramidal neurons* which we find mainly in layer V.

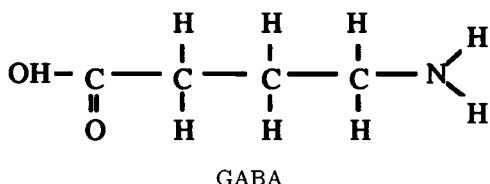
- Both families receive their input signals from *stellate cells*, which are interneurons lying in layer IV.
- The lower pyramidal neurons are output neurons. Their axons make contact with the thalamus.
- The upper neurons make distant connections with the pyramidal cells of other columns through the U-fibers.
- Within a column there are contacts between the pyramidal cells. Direct interpyramidal contact is excitatory. The neurotransmitter involved in the synaptic transmission is glutamate.



- The interneurons provide indirect inhibitory interactions between the pyramidal neurons. As a whole the effective interactions between pyramidal neurons can be of either sign, excitatory or inhibitory. Different

sorts of interneurons are involved, depending on the type of pyramidal neurons, upper or lower, to which they are related. For example, the basket cells connect the upper pyramidal cells and the chandelier the upper to the lower pyramidal cells. A model, devised by Sompolinsky *et al.*, shows how the interneurons could compel the average firing rate of pyramidal neurons to remain low (see section 12.5).

The neurotransmitter for inhibitory synapses is gamma-aminobutyric acid (GABA).



2.2.7 The organization of peripheral structures

- The hypothalamus and the reticular formation form a sort of primitive brain. The hypothalamus embeds fundamental programs associated with the survival of the individual or with the survival of the species (such as feeding, mating, breeding). The reticular formation carries out the programs by triggering stereotyped actions, chewing for example. The structure of the reticular formation shows few apparent regularities. It is made up of large bunches of crossing fibers, whence its name.
 - By contrast, the cerebellum, the nigrostriate formation and the hippocampus show well-defined, remarkably similar, organizations. This suggests that their various functions use the same type of information processes and that the basic functional principle of these structures is to carry out *comparisons*.
 - The *cerebellum* learns motor sequences. The motor cortex triggers an action. Then, through afferences coming from sensory organs, the actual situation is compared with the sequence. The comparison, according to Marr, occurs in the Purkinje cells, by the coactivation of two afferences, the signals coming from the sensory detectors through the climbing fibers on the one hand and the signals coming from associative motor cortex through the parallel fibers on the other. The Purkinje cells send the corrected signals to the motor cortex (Fig. 2.6.a).
 - The role of the *nigrostriate formation* seems to be that of allocating the motor activity according to the maps of the motor cortex. Brief and local activities, requiring the excitation of a specific region of the map, are controlled by this structure. However, large and long-lasting activities stimulate the secretion of a neurotransmitter, dopamine, which blocks the effects of globus pallidus, thus allowing numerous regions of the motor cortex to be excited simultaneously (see Fig. 2.6.b). The result of lesions of the nigrostriate formation is Parkinson's disease.
 - The *hippocampus*, a structure which we classify as peripheral, although not all neurobiologists would agree with this, is involved in long-term memory. Its destruction neither hinders the processes involved in short-term memory nor erases the items already stored in long-term memory, but it precludes the storage of any further information in the cortex. The hippocampus is thought to carry out spatial and temporal self-comparisons of the cortical activity: signals coming from the cortex

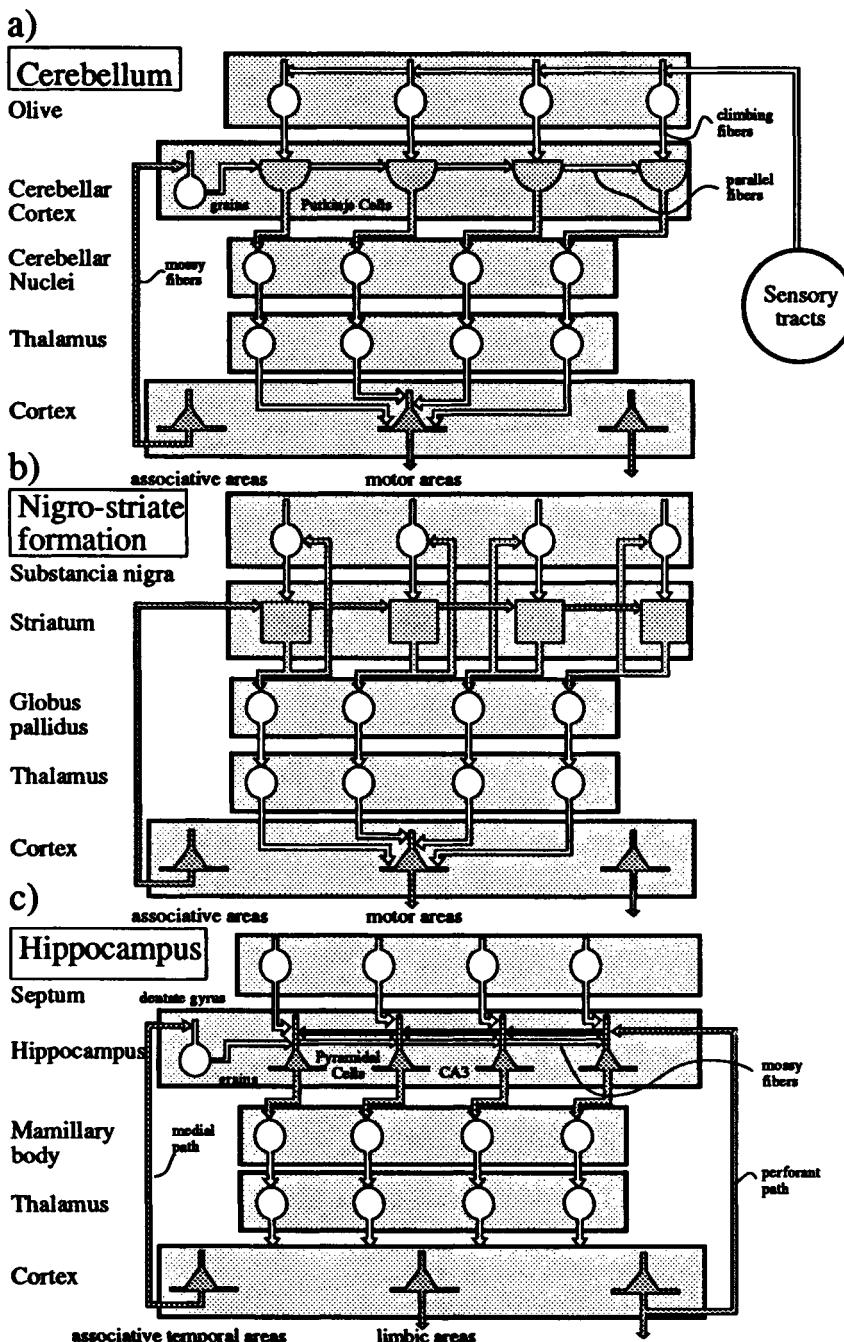
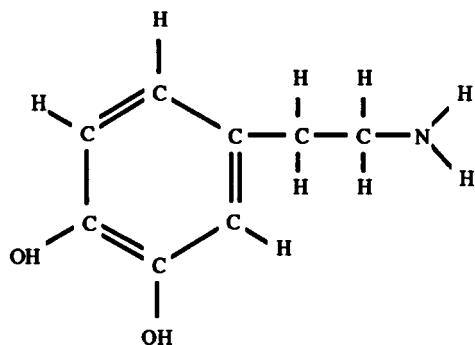
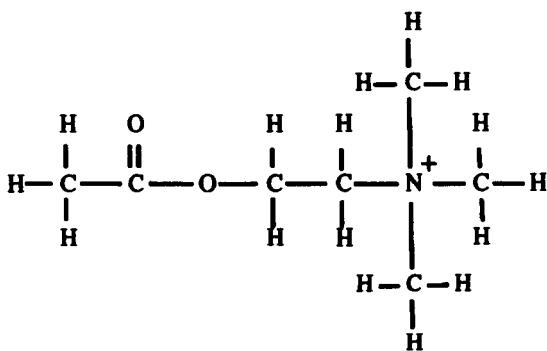


Figure 2.6. Schematic organization of three peripheral structures.



Dopamine

through two different pathways, one direct path via the perforant fibers and the other through the granule cells, coactivate the dendrites of the CA3 pyramidal cells. These cells project onto the limbic zones of the cortex. The hippocampus is also controlled by the septum and, therefore, by the fundamental programs of the hypothalamus: the patterns of activity close to those suiting the fundamental programs are more efficiently stored than the others (Fig. 2.6.c). The neurotransmitter of the septum is acetylcholine, the transmitter involved in neuromuscular junction.



Acetylcholine

One sees how complex is the organization of the CNS. However, the modeling considered in this text appeals only occasionally to the structures which have been described. Most models deal with homogeneous networks. They therefore apply to microcolumns where evidences for homogeneity are well documented. Extensions to columns or cortical bands are more risky. The theory of cortical tissue considered as an assembly of columns is still to be done.

2.3 A brief survey of neurophysiology

2.3.1 Action potential and neuronal state

The lipidic membrane of the neuronal cell is an obstacle that ions can only cross either through ionic pumps floating in the membrane or through channels, provided obviously that these channels are open. When the channels are closed the pumps create ionic concentration imbalances. The most common ions are Na^+ , K^+ and Cl^- . Their equilibrium concentrations are given in Table 2.1 (concentrations in mmol/lt.).

	Na^+	K^+	Cl^-
outside the cell	143	5	103
inside the cell	24	133	7

Table 2.1

The imbalance creates an electric potential through the membrane. According to Nernst's law the various contributions V_x of ions x to the membrane potential are given by

$$V_x = \frac{kT}{Z_x e} \log\left(\frac{n_{\text{out}}^x}{n_{\text{in}}^x}\right).$$

Z_x is the charge of ion x , T the temperature, n_{out}^x and n_{in}^x the ionic concentrations of x outside and inside the cell. The three contributions,

$$V_{\text{Na}^+} = +45 \text{ mV}, \quad V_{\text{K}^+} = -82 \text{ mV} \quad \text{and} \quad V_{\text{Cl}^-} = -67 \text{ mV},$$

combine and give a *resting membrane potential* of $V = -70 \text{ mV}$.

The sodium channels and the potassium channels are potential-dependant: a local electrical excitation causes the sodium channels to open. The sodium ions penetrate the cell and this motion locally depolarizes the membrane. The membrane potential increases up to about $+40 \text{ mV}$. This triggers the opening of the potassium channels at about 0.5 ms after the opening of the sodium channels. The potassium ions flow outside the membrane, thus repolarizing the membrane to -80 mV . The resulting peak of membrane potential, 1 ms long, is called an *action potential*. Afterwards, it takes about 3 to 4 ms for the channels to close and for the ionic pumps to reset the ionic concentrations (see Fig. 2.7). This resting time is called the *refractory period*.

The axon is an electric cable characterized by:

- its longitudinal ohmic resistivity, $R_0 = 50 \Omega \text{cm}$;
- its transverse ohmic resistivity, $R_m = 10^4 \Omega \text{cm}^2$, and

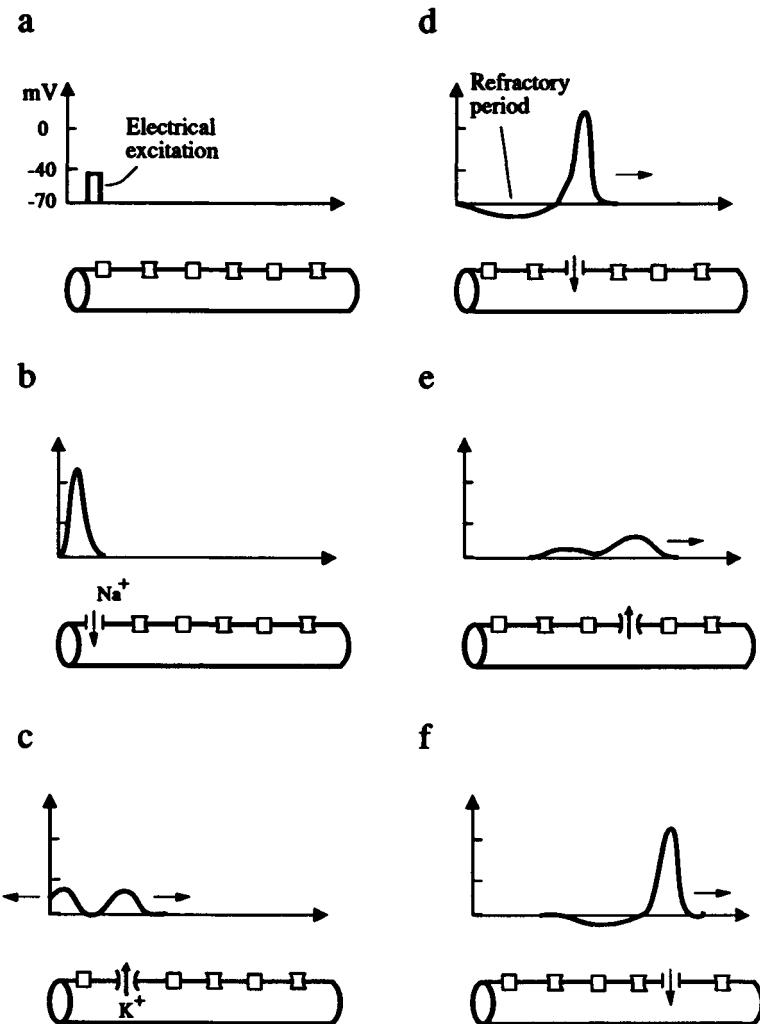


Figure 2.7. The creation and propagation of action potentials along the axonal membrane.

- its transverse capacitance $C_m = 1 \mu F cm^{-2}$.
These parameters determine the time constant,

$$\tau_m = R_m C_m \simeq 100 \text{ ms},$$

and the characteristic length of the cable,

$$\lambda = \frac{1}{2} \sqrt{\frac{dR_m}{R_0}} \simeq 0.07 \text{ cm},$$

for axons with a diameter of $d \simeq 1 \mu\text{m}$. The potential created by the opening of the ionic channels therefore diffuses according to the cable equation towards regions of the membrane, where it triggers the opening of new sodium and potassium channels. The cable equation, together with the dynamics of opening and closing of ionic channels, make up the Hodgkin Huxley equations. The process creates two waves, two action potentials, which propagate along the axonal membrane in opposite directions at about 5 ms^{-1} .

In actual fact, the initial excitation occurs at the root of the axon in a region, called the *hillock zone*, which contains a high density of channels (and also some special types of channels). This is the most sensitive part of the neuronal membrane, a region where an action potential is the most likely to burst. When the membrane potential of the hillock zone exceeds a *threshold* θ ($\theta \simeq -30 \text{ mV}$) the ionic channels open and an action potential propagates along the axon. The other action potential, however, cannot move in the other direction, towards the somatic and dendritic membranes. Indeed in those regions the channel density is too weak to trigger the propagation of action potentials and potentials can only diffuse.

Information in neurons therefore manifests itself in the form of electrical signals which propagate along the cell membrane according to two modes:

- A *propagative (or regenerative) mode in the axon*: the signal is a self-regenerating wave (called an *action potential*), the shape of which does not change as it moves along the axon. This is a logical signal.
- A *diffusive mode in the dendrites and in the soma*: the potential gradually damps out as it flows away from the excitation locus. The dendritic tree and the soma therefore carry analog signals.

Owing to its standard shape, an action potential does not embed any information except for that associated with its mere presence or absence. Owing to the fact that this is the sole information that a neuron sends to the other neurons, the state S_i of a neuron i can be considered as a *binary variable*

$$S_i(t) = 1$$

if the membrane potential in the hillock zone of neuron i at time t is not the resting potential and

$$S_i(t) = 0$$

if the membrane potential at time t takes its resting value.

We call τ_r the firing duration ($\tau_r \simeq 4$ ms). It includes both the firing period and the refractory period which make up an action potential.

It is sometimes convenient to introduce *symmetrical variables* defined by

$$\sigma_i(t) = \begin{cases} 1 & \text{if the neuron is firing at time } t, \\ -1 & \text{if it is silent.} \end{cases}$$

The two definitions of activity are related by

$$\sigma_i = 2S_i - 1. \quad (2.1)$$

The firing rate, or ‘instantaneous frequency’, is defined by the number of spikes (of action potentials) emitted per unit time. If $n_i(t, \Delta t)$ is the number of spikes sent out by the neuron i during Δt around t , the instantaneous frequency of i is given by

$$\omega_i(t) = \frac{n_i(t, \Delta t)}{\Delta t}.$$

The instantaneous frequency is related to activities $\overline{S_i(t)}$ and $\overline{\sigma_i(t)}$, averaged over time windows Δt , by the following formula:

$$\overline{S_i(t)} = \frac{1}{2}[1 + \overline{\sigma_i(t)}] = \tau_r \omega_i(t). \quad (2.2)$$

The maximum firing rate is achieved when

$$\overline{\overline{S_i}} = \overline{\overline{\sigma_i}} = 1.$$

Then

$$\omega_i = \omega_{\max} = \frac{1}{\tau_r}.$$

Saltatory propagation

For the sake of survival it is important for the nervous system to carry the electrical signals as fast as possible. This is achieved in molluscs by increasing the diameters of the axons. Indeed, the propagation velocity scales as the square root of the diameter of the fiber. However, this process is not too efficient. A cleverer trick is the saltatory propagation. Ancillary glial cells cover the axons with lipidic sheets or myelin. Myelin, because it is a poor conductor, improves the cable properties of the axon to a considerable extent. In particular, its characteristic diffusion length is much larger and its characteristic time is much shorter. Myelin is a barrier for ions and the membrane of a myelinated axon does not comprise any ionic channel. Propagation in myelinated axons is only diffusive. But myelin does not cover the whole surface of the axon. From place to place, about every millimeter, at loci called Ranvier nodes, the axon is free of myelin and the channel density is very high. The axon therefore fires at a Ranvier node, and the potential diffuses to the next distant

Ranvier node where the high channel density and the enlarged diffusion length make it possible for the axon to fire again. This is called saltatory propagation. Owing to the shortening of the time constant the propagation velocity associated to this mode increases up to 100 ms^{-1} .

2.3.2 Synaptic transmission and somatic integration

The synaptic buttons lying at the tips of the branches of the axonal tree embody vesicles which contain the neurotransmitter molecules. An action potential propagating along the button membrane opens Ca^{2+} channels lying in the membrane and these ions in turn activate the fusion of the membrane of the vesicles and that of the button itself (exocytosis) (see Fig. 2.8). The neurotransmitter molecules then flow into the synaptic cleft (about 500 \AA wide). These molecules diffuse until they eventually get fixed on large proteins, the neuroreceptors attached to the membrane of the postsynaptic neuron. This reaction triggers a geometric (allosteric) transformation of neuroreceptors, which enables specific ions to flow inside the postsynaptic membrane. When the ions are Na^+ ions the synapse is excitatory: the flow creates a small depolarizing potential of about $100 \mu\text{V}$. When the ions are Cl^- ions the synapse is inhibitory: this ionic flow creates a small hyperpolarizing potential.

The transformation of the action potential into a postsynaptic potential lasts about 1 ms. A postsynaptic potential diffuses along the membrane of the dendrites and that of the soma till it reaches the hillock zone of the neuron. There it takes the form of a smooth wave,

$$J_{ij} \chi(t),$$

where $\chi(t)$, the ‘synaptic memory’, is a standard response function to an impinging action potential. The function is normalized,

$$\int dt \chi(t) = 1,$$

and its width is characterized by a time τ_p (which eventually depends on the location of the synapse on the dendrite). J_{ij} is the amplitude of the response of the membrane potential at the hillock zone of neuron i due to the emission of an action potential by neuron j . It is called the *synaptic efficacy* of the synapse connecting j to i . Second-order details having been set aside, the postsynaptic potential induced by the activity of j on i at time t is given by

$$v_{ij}(t) = J_{ij} \int_0^\infty d\tau \chi(\tau) S_j(t - \tau_{ij} - \tau), \quad (2.3)$$

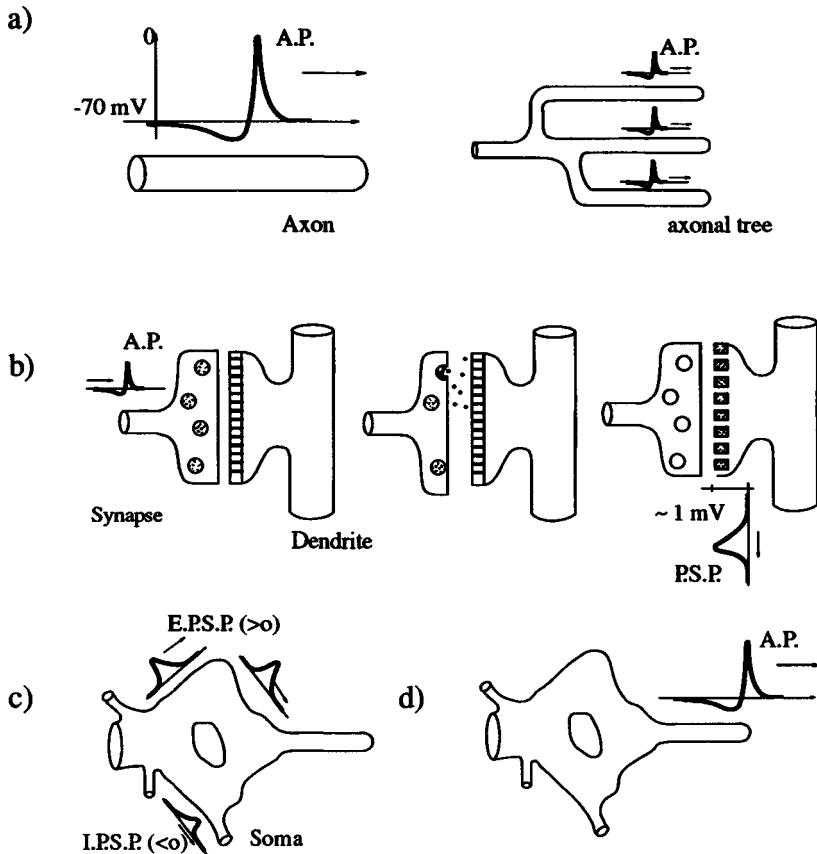


Figure 2.8. The various steps of neuronal dynamics.

- a) Axonal propagation.
- b) Synaptic transmission.
- c) Somatic integration.
- d) The triggering of a new action potential.

where τ_{ij} is the transmission delay between the hillock zone of neuron j and that of neuron i .

The existence of a synaptic memory makes the theory of neural networks very difficult, even for ‘networks’ reduced to a single neuron. The dynamics of self-connected single neurons with synaptic memory effects has been studied by Cosnard. It is discussed in section 3.1.2. Simplifications are then necessary. One recalls that the existence of a refractory period τ_r compels the dynamical variables S_j to keep the value $S_j = 1$ for times greater than τ_r . If $\tau_p < \tau_r$ the expression (2.3)

can be replaced by

$$v_{ij}(t) = J_{ij} S_j(t - \tau_{ij}). \quad (2.4)$$

This is in general an acceptable approximation which makes the analysis much easier. For example, it allows the treating of excitatory and inhibitory synapses on the same footing. The inhibitory postsynaptic potentials are shallower and last longer than the excitatory postsynaptic potentials. Therefore one would have to distinguish between two sorts of response functions χ^+ and χ^- . In the limit of short enough τ_p' s, however, the distinction vanishes and the postsynaptic potentials are given by Eq. (2.4) in both cases. Only the sign of synaptic efficacies J_{ij} differs. It must also be stressed that the injection of certain neurotransmitters such as serotonin stretches the duration of the postsynaptic potentials, that is to say τ_r gets larger. This obviously increases the amplitude of the efficacy J_{ij} , but the temporal effects of such a modification could also be of biological significance.

On the other hand, *the synaptic transmission is not a deterministic process*. The opening of one vesicle is called a quantum. Generally an action potential activates one quantum at most and very often it fails to activate any quantum at all. The probability of an action potential triggering the opening of one vesicle is $p \simeq 0.6$. This important phenomenon is the main source of noise in neural networks. The postsynaptic potential induced by j on i at time t is therefore given by

$$v_{ij} = \begin{cases} J_{ij} S_j(t - \tau_{ij}) & \text{with probability } p, \\ 0 & \text{with probability } (1 - p). \end{cases} \quad (2.5)$$

Observation of synaptic quanta

The stochastic nature of synaptic transmission has been studied by Katz in particular. One observes the amplitude of the postsynaptic potential elicited at a neuromuscular junction by the excitation of the afferent fiber. In general there exist several synaptic contacts between the fiber and the postsynaptic membrane. It is possible to determine the number of these synapses by using microscopy techniques. The experimental curve shows a series of peaks. It can be perfectly analyzed by assuming that:

- the action potential triggers the opening of one vesicle at most at each synapse;
- the opening is a stochastic process obeying a Poisson distribution with an average probability $p = 0.6$ that the vesicle indeed releases its neurotransmitter content into the synaptic cleft.

The peaks correspond to the simultaneous release of one quantum by one, two, ... synapses (see Fig. 2.9).

Finally, the postsynaptic potentials add up algebraically in the hillock zone of neuron i , a phenomenon called *somatic integration*. The result of this summation is a membrane potential given by

$$v_i = \sum_j v_{ij}. \quad (2.6)$$

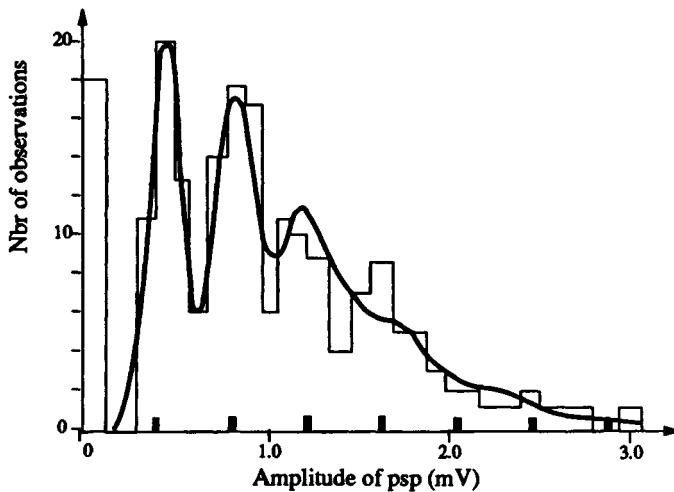


Figure 2.9. Histogram of postsynaptic potentials observed at a muscular junction (After Katz). The junction comprises several synapses and the peaks correspond to the simultaneous excitation of 0, 1, 2, ... synapses. The best fit yields a synaptic transmission probability of $p = 0.6$ p.s.p per AP.

This potential is to be compared with the threshold θ_i :

$$S_i(t) = \begin{cases} 1 & \text{if } t - t_i < \tau_r \text{ or } v_i(t) > \theta_i, \\ 0 & \text{if } t - t_i \geq \tau_r \text{ and } v_i(t) \leq \theta_i, \end{cases} \quad (2.7)$$

where t_i is the moment the neuron i was updated from $S_i = 0$ to $S_i = 1$ for the last time. (This is the blocking effect due to the existence of the refractory period.)

Remark

Two neurons can be connected by more than one synapse. The synaptic efficacy J_{ij} actually stands for all these physical synapses. This is an effective synapse. We have seen that the cortical pyramidal cells are linked either directly through excitatory synapses or indirectly through inhibitory interneurons. The effective synapse between pyramidal neurons can therefore be of either sign. Anatomical observations show however that the axonal branches of pyramidal neurons tend to stretch along straight lines with the consequence that monosynaptic contacts between pyramidal neurons seem to be the rule and polysynaptic links the exception (Braitenberg).

2.3.3 Dealing with noise

We have seen that the postsynaptic potentials are stochastic variables. Since the membrane potential v_i is a sum of postsynaptic potentials it is also a stochastic variable and because it is the sum of a large number of independent stochastic variables its distribution is Gaussian (the

central limit theorem). One recalls that a Poisson distribution becomes a Gaussian distribution when the number of events increases. The distribution is determined by its average and its mean square deviation.

- The mean value $\overline{v_i(t)}$ of the *membrane potential* of neuron i at time t is given by

$$\overline{v_i(t)} = \sum_j \overline{v_{ij}(t)} = p \sum_j J_{ij} S_j(t - \tau_{ij}) \quad (2.8)$$

since $\overline{v_{ij}(t)} = \int dv_{ij} v_{ij} P(v_{ij}) = p J_{ij} S_j(t - \tau_{ij})$.

This average may be considered either as the most probable value of the membrane potential of the neuron i of a given neuronal network or as an average over the actual values of membrane potentials measured in a set of identical networks on neurons with the same label i (an ensemble average).

- Similarly, we compute the mean square deviation $\overline{\Delta v_i^2}$ of the potential v_i :

$$\begin{aligned} \overline{\Delta v_i^2} &= \overline{v_i^2} - (\overline{v_i})^2 = \sum_j \sum_k \overline{v_{ij} v_{ik}} - \sum_j \overline{v_{ij}} \sum_k \overline{v_{ik}} \\ &= \sum_j \overline{v_{ij}^2} - \sum_j (\overline{v_{ij}})^2 + \sum_j \sum_{k \neq j} \overline{v_{ij} v_{ik}} - \sum_j \sum_{k \neq j} \overline{v_{ij}} \overline{v_{ik}}. \end{aligned}$$

Using $\overline{v_{ij}^2} = \int dv_{ij} v_{ij}^2 P(v_{ij}) = p J_{ij}^2 S_j(t - \tau_{ij})$, since $S_j^2 = S_j$, one finds:

$$\sum_j \overline{v_{ij}^2} = p \sum_j J_{ij}^2 S_j(t - \tau_{ij}).$$

Also, $\sum_j (\overline{v_{ij}})^2 = p^2 \sum_j J_{ij}^2 S_j(t - \tau_{ij})$. On the other hand,

$$\overline{v_{ij} v_{ik}} = \iint dv_{ij} dv_{ik} v_{ij} v_{ik} P(v_{ij}, v_{ik}).$$

Since the synaptic transmission probabilities are independent variables, one can write

$$\overline{v_{ij} v_{ik}} = \int dv_{ij} v_{ij} P(v_{ij}) \int dv_{ik} v_{ik} P(v_{ik}) = \overline{v_{ij}} \overline{v_{ik}}$$

and therefore

$$\sum_j \sum_{k \neq j} \overline{v_{ik} v_{ik}} = \sum_j \sum_{k \neq j} \overline{v_{ij}} \overline{v_{ik}}.$$

The expression of the mean square deviation then reduces to

$$\overline{\Delta v_i^2} = p(1-p) \sum_j J_{ij}^2 S_j(t - \tau_{ij}). \quad (2.9)$$

Assuming that the activity S_j and the square of synaptic efficacies J_{ij} are uncorrelated quantities, Eq. (2.9) is written as

$$\overline{\Delta v_i^2} = p(1-p) z \Delta J_i^2 \langle S \rangle,$$

where $\langle S \rangle$ is the average neuronal activity of the network, ΔJ_i^2 is the mean square synaptic efficacies of synapses impinging on neuron i ,

$$\Delta J_i^2 = \frac{1}{z} \sum_j J_{ij}^2,$$

and z is its connectivity, that is, the average number of neurons a given neuron is linked to.

$\overline{\Delta v_i^2}$ is a local noise. In the forthcoming developments we assume that the distribution of synaptic efficacies is homogeneous and we define a noise parameter B :

$$\overline{\Delta v_i^2} = B^2 = p(1-p) z \Delta J^2 \langle S \rangle, \quad (2.10)$$

$$\text{with } \Delta J^2 = \frac{1}{N} \sum_i \Delta J_i^2.$$

The noise parameter B depends on the average activity. In most of the models we consider in this text the average activity is $\langle S \rangle = 0.5$ but the average activity of actual systems is considerably lower, of the order of 0.1. Although we shall consider noise as a constant parameter, it could be worth considering the effect of noise reduction when dealing with systems designed to show low activities.

- The *relative membrane potential* (or *local field*, to follow the term used in statistical physics) is defined as

$$h_i = v_i - \theta_i. \quad (2.11)$$

Its distribution is given by

$$P(h_i) = \frac{1}{\sqrt{2\pi B^2}} \exp\left[-\frac{(h_i - \bar{h}_i)^2}{2B^2}\right], \quad (2.12)$$

where, according to Eq. (2.8), \bar{h}_i the local mean field, is given by

$$\bar{h}_i = \sum_j p J_{ij} S_j(t - \tau_{ij}) - \theta_i.$$

One observes that the existence of a synaptic noise amounts to replacing J_{ij} by pJ_{ij} in the expression of the local mean field. For the sake of simplicity this factor is made implicit in all forthcoming developments.

The probability for the local field to be positive is given by

$$P[h_i > 0] = \int_0^\infty dh_i P(h_i).$$

Introducing the error function $\text{erf}(x)$ defined by

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x dt \exp(-t^2), \quad (2.13)$$

the probability becomes

$$P[h_i > 0] = \frac{1}{2} \left[1 + \text{erf}\left(\frac{\bar{h}_i}{B\sqrt{2}}\right) \right]. \quad (2.14)$$

- To summarize, the dynamics of biological neurons rests on a set of basic mechanisms which are retained in the building of a theory of formal neurons. These are:

- The nature of the transmission of information between the units of the network through standard signals, the action potential. This observation allows the consideration of formal neurons as binary automata.
- The existence of a refractory period, which provides a standard of time to the formal network.
- The way information is processed through excitatory or inhibitory synapses, which is modeled in efficacies of both signs.
- The mechanism of somatic integration, which manifests itself in the form of a local field on the units of the formal system.
- The threshold mechanism, which makes the neuronal systems members of the family of threshold automata networks.
- Finally, the probabilistic nature of the synaptic transmission, which is summed up in the formulae

$$P[S_i(t) = 1] = \begin{cases} 1 & \text{if } t - t_i < \tau_r, \\ 1 - P[S_i(t) = 0] = P[h_i > 0] & \text{otherwise,} \end{cases} \quad (2.15)$$

(where t_i is the time of the last updating to $S_i = 1$) compels us to regard the neural systems more precisely as probabilistic threshold automata

networks. The study of the dynamics defined in 2.15 is the subject of Chapter 3.

2.3.4 Refinements of the model

- *Imperfect transmissions.* — When arriving at a branching of the axonal tree the action potential must divide along all branches of the tree. But it can fail to do so. In actual fact there exists a probability of the action potential not continuing its way along a given branch; the thinner the branch, the higher the probability. This effect can be modeled by the probability that an action potential sent out by a neuron j arrives at the synaptic button (ij) . Instead of a single probability p one has now to deal with probabilities p_{ij} . This simply changes the definition of the local mean field and that of local noise:

$$\bar{h}_i = \sum_j p_{ij} J_{ij} S_j(t - \tau_{ij}) - \theta_i,$$

with

$$B_i^2 = \langle S \rangle \sum_j p_{ij} (1 - p_{ij}) J_{ij}^2.$$

- *Noisy threshold.* — The threshold is also a fluctuating parameter. This source of noise adds its effect to that of synapses in the fluctuations of the local field. Because the sources are independent, the overall noise is given by

$$B^2 = (B_{\text{syn}})^2 + (B_{\text{thres}})^2.$$

It is worth noting that the effects of fluctuating thresholds and those of probabilistic synapses are alike because both induce fluctuations of local fields. In the simulations or in the building of neuronal machines it is much easier to make threshold fluctuate, and this is the way noise is introduced in artificial neuronal devices (see section 11.1.1).

- *Heterosynaptic junctions.* — Up to now we have considered binary synapses only: one neuron is connected to another neuron through a single synaptic cleft. This is a homosynaptic (axo-dendritic) junction. The local field is a linear function of incoming activities.

Kandel pointed out that other types of junctions could exist. For example a neuron k can make a contact on a synapse (ij) itself (an axo-axonic contact, see Fig. 2.10.a). The activity of this neuron k modifies the postsynaptic potential created by the main synapse according to

$$v_{ij} = \begin{cases} a S_j & \text{if } S_k = 0, \\ b S_j & \text{if } S_k = 1. \end{cases}$$

These expressions can be lumped together, $v_{ij} = a S_j + (b - a) S_j S_k$ or

$$v_{ij} = \sum_k (a S_j + (b - a) S_j S_k),$$

if there are several neurons k controlling the synapse (ij) .

The membrane potential becomes

$$\bar{h}_i = \sum_j J_{ij} S_j(t - \tau_{ij}) + \sum_j \sum_k J_{ijk} S_j(t - \tau_{ij}) S_k(t - \tau_{ik}) - \theta_i.$$

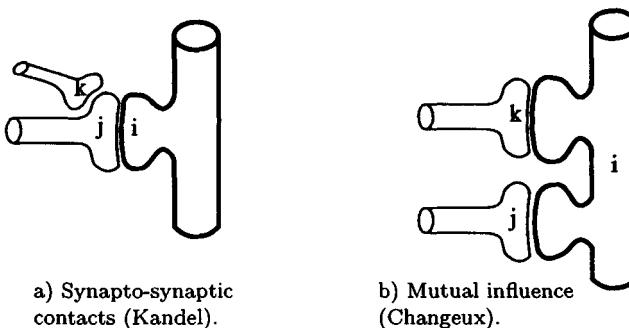


Figure 2.10. Two models of heterosynaptic junctions.

Therefore the overall connection is third-order, involving synapses (ijk), and the local field is a quadratic function of the activities. These connections are called heterosynaptic junctions. Axo-axonic contact is one among other hetero-junction mechanisms. For example, it has been shown by Changeux that two nearby synapses could influence each other (Fig. 2.10.b). Applications by Nadal, Dehaene and Changeux of these ideas to the mechanisms by which birds learn songs are discussed in section 5.2.5. The argument is identical to the one developed for axo-axonic contacts, except that the synaptic efficacies are more symmetrical:

$$J_{ijk} \simeq J_{ikj}.$$

Such hetero-junctions have been evoked in the functioning of the cerebellum and that of the hippocampus (see section 10.5). In the cortex the electron micrographs show such an entanglement of fibers that crossed influences are likely in this organ as well.

- *Gap junctions.* — Gap junctions (or electrical synapses) are physical contacts between the membranes of presynaptic and postsynaptic neurons. No chemical transmitters are involved. As in chemical synapses, gap junctions give rise to postsynaptic potentials and therefore they can be described by the same formalism. The difference is that they are fast processes, since they do not involve molecular diffusion mechanisms, and that they are non-plastic, which means that they cannot be implied in learning processes.

2.4 Learning and memory: a summary of experimental observations

The last section dealt with the modeling of neuronal states dynamics. Living organisms, however, are adaptive systems: the responses of their neuronal system must adapt themselves to environmental constraints and this can be achieved only through modifications of parameters which determine the network, the synaptic efficacies in particular. Therefore, besides the neuronal dynamics, there must exist learning dynamics. Both dynamics have to be coupled to each other. It is of paramount importance to check whether observation indeed supports these hypotheses,

whether real neural networks are adaptive, and, if the case arises, how plasticity is achieved. This section is devoted to those results of experimental psychology and neurophysiology which, we think, are relevant for the modeling of learning.

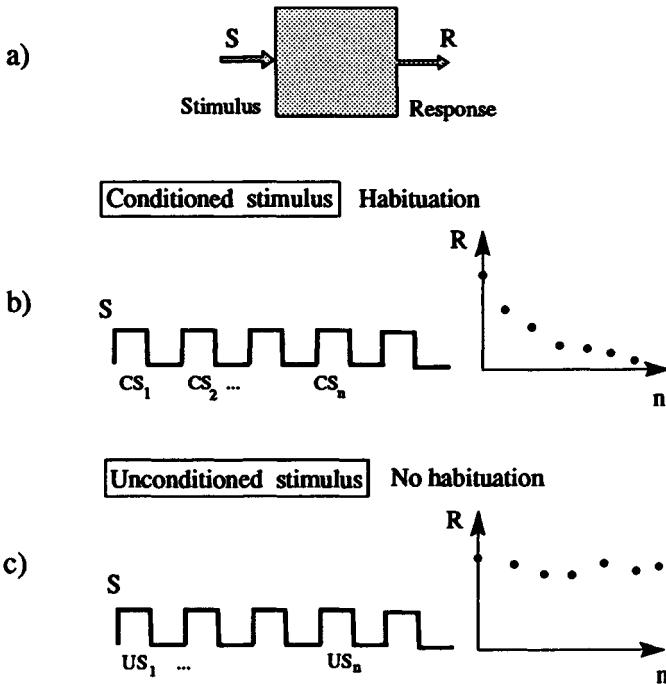


Figure 2.11. a) The black box and its two types of responses: b) to conditioned stimuli (habituation); c) to unconditioned stimuli.

2.4.1 Classical conditioning

Many experiments in physics are carried out by looking at the properties of samples experiencing well-defined excitations. The very same protocol is used by the behaviorists to work on living material. According to their terminology, an excitation is called a stimulus S . The output triggered by S is called the response R .

a) *Habituation.* — In the simplest experiments, one measures the amplitude of the response R according to the number of stimuli undergone by the studied organism. In some experiments the amplitude of the response decreases when the number of stimuli increases (Fig. 2.11.a). This phenomenon is called habituation and the stimulus which gives rise

to habituation is called a conditioned stimulus (CS). For example the sound of a bell (CS) makes a dog turn his head away. After several such stimulations the dog does not turn his head any more (Fig. 2.11.b).

Other stimulations do not give rise to habituation: the response remains the same whatever the number of stimuli. These stimuli are called unconditioned stimuli (US) and the associated response is called the unconditioned response (UR) (Fig. 2.11.c). The sight of meat (US), for example, always makes a hungry dog salivate (UR).

b) *Classical conditioning.* — An animal is trained with two stimuli, an unconditioned stimulus (US) and a conditioned stimulus (CS). One observes the unconditioned response (UR) associated with the (US). At the beginning of the experiment the conditioned stimulus (CS) does not prompt any response. But after a number of trainings, it becomes able to elicit the response by itself (Fig. 2.12). In Pavlov's classical experiment a hungry dog simultaneously experiences the sound of a bell (CS) and the sight of meat (US). After some training the dog salivates at the sole sound of the bell.

More refined experiments show that the CS is to be experienced *before* the US, for the conditioning to be efficient. Actually it is the most efficient when the inter-stimuli interval (ISI) is about 0.2 s.

A model (called Limax) of classical conditioning by a neural network reminiscent of that of a slug has been put forward by Gelperin and Hopfield. It is discussed in section 5.3.

c) *Compound conditionings and context effects*

- *Serial stimuli* (and responses, cf. Fig. 2.13.a). — Once a stimulus CS1 elicits a response UR it can be considered as an unconditioned stimulus for UR. It is therefore possible to associate CS1 with another stimulus CS2 and CS2 triggers UR in turn. In this way an animal is able to learn full chains of stimuli and therefore the chains of associated unconditioned responses. For behaviorists this is the basic mechanism for behavioral learning in animals.

- *Parallel responses* (context effects) (Fig. 2.13.b). — In this experiment the system undergoes simultaneously several conditioned stimuli together with the unconditioned stimulus. This is actually an unavoidable situation since it is difficult to isolate the wanted, specific stimulus from other environmental stimuli. One observes that the presence of context stimuli weakens the response to the specific stimulus.

2.4.2 Modeling of classical conditioning

The responses generally show up non-linear effects. Their amplitudes S^r are sigmoidal functions of the strengths S^s of stimuli. The response

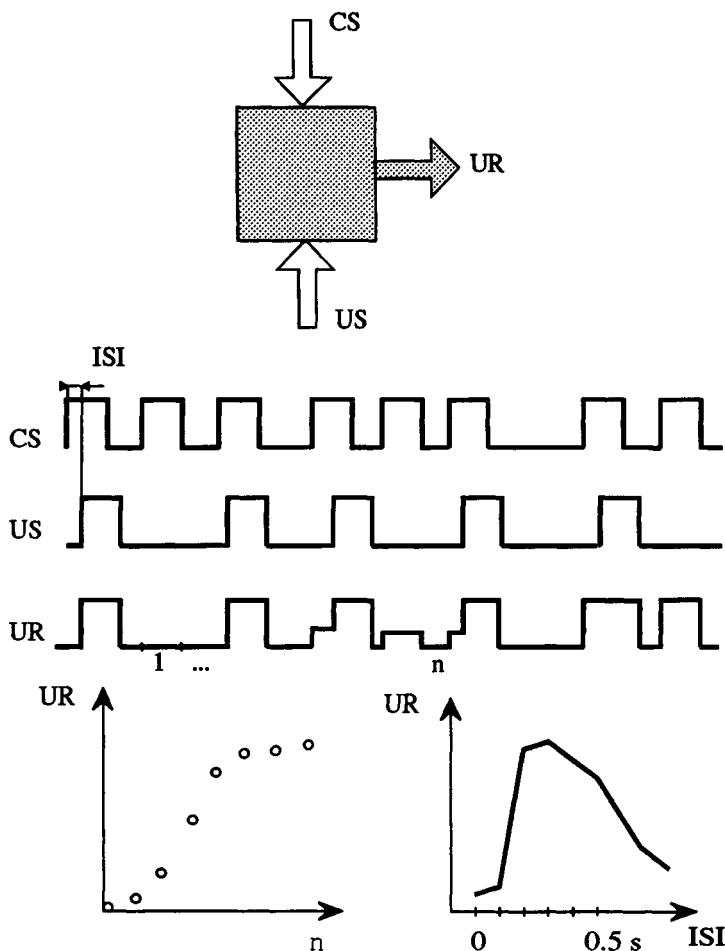


Figure 2.12. A classical conditioning experiment.
 CS: conditioned stimulus; US: unconditioned stimulus;
 UR: unconditioned response; ISI: interstimulus interval.

function can be crudely modeled by a step function which depends on a threshold value θ :

$$S^r = \mathbf{1}(JS^s - \theta) \quad \text{with} \quad \mathbf{1}(x) = \begin{cases} 1 & \text{if } x > 0, \\ 0 & \text{otherwise,} \end{cases} \quad (2.16)$$

where J is a parameter which describes how large the influence of the stimulus S^s is in eliciting the response S^r . We take $S^s = 1$ if the stimulus is present, $S^s = 0$ otherwise. When several stimuli are present one simply

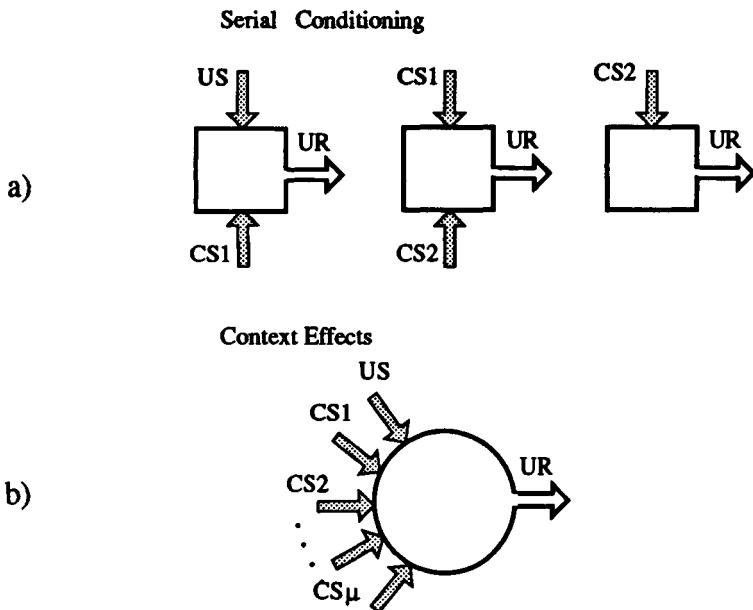


Figure 2.13. Serial and parallel conditioning experiments.

assumes that their influences sum up:

$$S^r = \mathbf{1} \left(\sum_s J^s S^s - \theta \right). \quad (2.17)$$

The equations (2.17) are similar to those driving the dynamics of a noiseless neural network.

In classical conditioning experiments one has

$$S^r = \mathbf{1} (J^u S^u + J^c S^c - \theta), \quad (2.18)$$

where S^u is the unconditioned stimulus and S^c the conditioned stimulus. Before the starting of the training session we have $J^c < \theta$ for a conditioned stimulus ($S^r = 0$) and $J^u > \theta$ for an unconditioned stimulus ($S^r = 1$).

During the training the influence J^u of the US does not change but the influence J^c of the CS is assumed to be modified when the CS and the UR are both active. This learning rule can be written as

$$\Delta J^c = \varepsilon S^r S^c, \quad \text{with } \varepsilon > 0. \quad (2.19)$$

It accounts for the main properties of classical conditioning, in particular for the sigmoid learning curve displayed in Fig. 2.11 (with the step

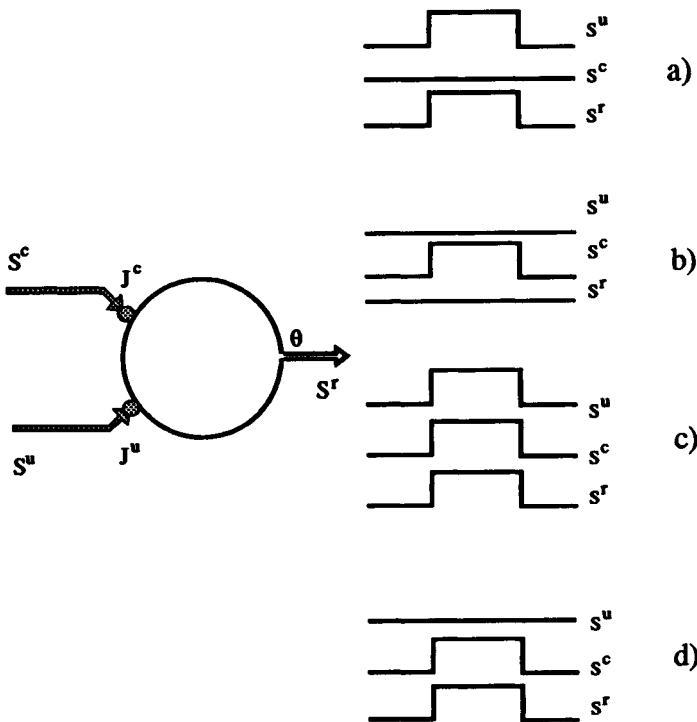


Figure 2.14. Formalization of classical conditioning experiments.

function $\mathbf{1}(x)$ of Eq. (2.18) replaced by a sigmoidal function), but not for more subtle effects such as the delay effects.

The similarity between the equations determining the responses of the system as a whole and the equations driving the dynamics of neurons suggested to Hebb that the learning rule given in Eq. (2.19) could apply at the neuronal level. The synaptic efficacy is the analog of influence J and the pre- and postsynaptic neuron activities are identified with the stimulus and response states respectively:

$$\Delta J_{ij} = \varepsilon S_i S_j, \quad (2.20)$$

this is known as the Hebbian rule. j is the label of the upstream neuron and i is that of the downstream neuron. Hebbian models are studied in Chapter 4.

Context effects can also be modeled: the negative effect of a context stimulus is proportional to its influence J^s and therefore to

$$-J^s S^s S^c.$$

Adding all the contributions we obtain a modified learning rule:

$$\Delta J^c = \varepsilon \left(S^r - \lambda \sum_{s \neq c} J^s S^s \right) S^c, \quad \lambda > 0; \quad (2.21)$$

this is due to Rescorla and Wagner.

Application of the Rescorla Wagner rule to neurons leads to the formula

$$\Delta J_{ij} = \varepsilon \left(S_i - \lambda \sum_{k \neq j} J_{ik} S_k \right) S_j, \quad (2.22)$$

a rule which is due to Widrow and Hoff. Learning mechanisms which use this rule are discussed in sections 7.3.1 and 7.3.3.

2.4.3 Instrumental (or operant) conditioning

Classical conditioning is a passive process. It consists of the substitution of a fundamental program by an imposed one. Instrumental conditioning, on the contrary, is an active conditioning: the animal has to modify its behavior to satisfy a constraint imposed on it by the experimenter. It must find the optimal behavior by itself. The animal moves in a rigorously controlled, standard environment such as a maze or a Skinner box. It is rewarded, with some appetizing food, if it succeeds in doing a given task. This protocol is called a positive reinforcement. Negative reinforcement, delivering electric shocks to the animal for example, is also used.

In instrumental conditioning the animal must anticipate the outcome of its behavior. The explanation, in terms of neural networks, of these experiments is certainly difficult. To show that instrumental conditioning brings large neural networks into play, it is enough to observe that it discriminates between animals comprising millions of neurons. Let us give two examples.

- *The T maze.* — This simple maze has only two branches, branch A and branch B. Food is to be found at the tip of one of the branches. Most ‘higher’ animals quickly learn to go to the right branch. Let us now introduce a bit of randomness. Let us say food is placed on branch A with probability 0.8 and on branch B with probability 0.2. Animals ‘up’ to the rat adapt their strategy to the probability: they turn to branch A with probability 0.8 and to branch B with probability 0.2. Cats always go to branch A. They are right, since their probability of finding food is 0.8 whereas the probability for rats is $0.8 \times 0.8 + 0.2 \times 0.2 = 0.68$.

- *The learning sets.* — A monkey is presented with two boxes. The boxes have different characteristics, a different form, square and round for example, and they are placed at random in front of the monkey. One

of the boxes, the square one for example, conceals a bit of a food the monkey is very fond of. It takes about ten trials for the monkey to learn that the food is hidden under the square box. Then the experiment is repeated with a rectangular box and a triangular box. The monkey learns that the food is under the triangular box in eight trials. After a set of about one hundred such series of experiments (the learning set) the monkey understands that at most two trials are enough to decide which box is the right one (Fig. 2.15.a).

The same experiment is carried out with different species (see Fig. 2.15.b). It discriminates very well between the species and even between the families of a same species. Monkeys are ‘cleverer’ than cats and cats are cleverer than rats, which seems to follow common sense. However, this conclusion depends heavily on the learning protocol and one must be extremely cautious when attributing degrees of intelligence between the various species: for example, rats are superior to monkeys if the experiment appeals to olfactory rather than visual stimuli.

The following are the main regularities brought about by operant conditioning experiments:

- *Temporal relationship*: an efficient behavior must be immediately positively reinforced.
- *Extinction*: an efficient behavior is forgotten if it is not reinforced.
- *Generalization*: a behavior which is efficient with respect to a certain environment is still adapted to environments which are not too perturbed. For example, if an animal is trained to respond to a given tune, it also responds to sounds whose frequencies are close to the training frequency.
- *Discrimination*: animals are generally able to discriminate between two very close environments if the response to the first environment is positively reinforced and that to the other environment is negatively reinforced. For example, animals can be trained to respond differently to two sounds whose frequencies are very close one another.

2.4.4 Memory: experimental psychology observations

- *Short-term and long-term memories*. — Observation of memorization phenomena in man have shown the coexistence of two types of memory: an erasable short-term memory and a quasi-irreversible long-term memory. The observation implies two series of experiments. In the first one devised by Murdock the subject has to learn a string of words. Then he has to retrieve them. The percentage of correct retrievals of a word is plotted along the position of the word in the string. The very first words are the most faithfully retrieved (initiality effect). So are the last memorized words (recency effect). Postman and Phillips carried out the

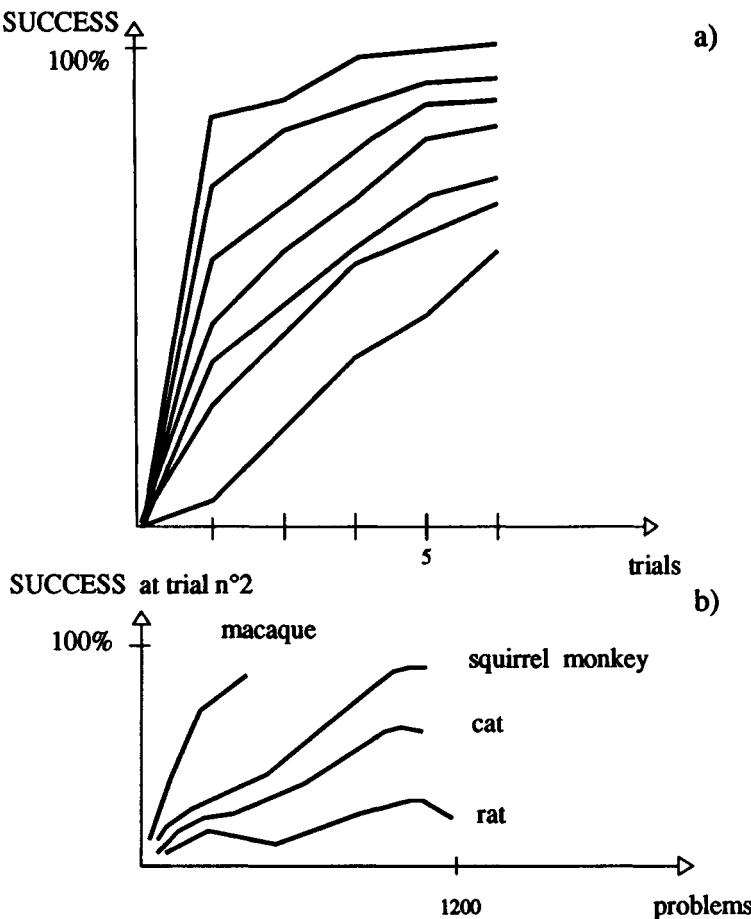


Figure 2.15. Learning sets experiments.

a) Carried out on monkeys. Each curve is an average of over 20 problems. The performance increases as the number of problems increases (After Hall).

b) Carried out on different species (After Macintosh).

same type of experiment, except that the subject, once he had learned the string, was asked to do a different task such as counting backwards starting from a given number. This extra task erases the recency effect, which is therefore attributed to short-term memory (see Fig. 2.16).

Short-term memory lasts a few seconds and allows the storage of about seven items. This number does not depend on the nature of the items provided they are not correlated. The items, for example, can be seven

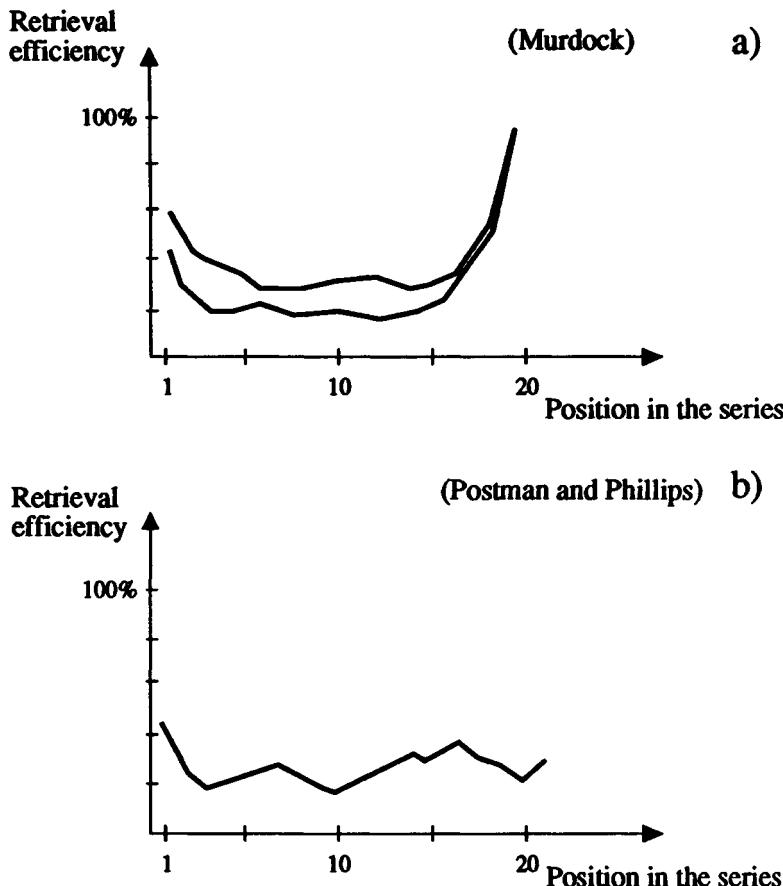


Figure 2.16. Short- and long-term memories, as observed in a retrieval task. The two experiments, that of Murdock and that of Postman and Phillips, are identical except that in the latter an extra task is carried out by the subject at the end of the storage session which erases the short-term memory effect.

letters not forming a word or they can be seven words not making a sentence, etc. A theory which possibly accounts for this result has been put forward by Nadal and Toulouse. It is discussed in section 7.2.2.

The *long-term memory* process needs some time, a consolidation period of a few seconds, to be efficient. This can be seen in Fig. 2.16: the percentage of correct retrieval increases when the items are presented with a lower frequency. It has also been observed that the memorized patterns organize themselves over long periods of time, up to several days.

- *Recall and retrieval.* — The experiments so far described are *retrieval* experiments. Since they do not involve any external stimulus, they cannot be carried out on animals. In *recall* experiments the subject is asked to recognize objects: he has to respond positively to stimuli belonging to the series of learned objects and to ignore stimuli not belonging to the series. Recall is easier than retrieval. It can be applied to animals but it is not certain that both protocols put the same mechanisms into play.

- *Semantic memory.* — A subject has to say whether propositions of type ' $X = Y_i$ ' are true or false for various Y_i . The time t which it takes to give a correct answer is measured.

It is assumed that this time depends on the semantic distance between X and Y_i ; the larger the time, the greater the distance. The experimental results obtained on classical examples such as

$$\left\{ \begin{array}{ll} \text{a canary is a canary} & \text{time } t_1, \\ \text{a canary is a bird} & \text{time } t_2, \\ \text{a canary is an animal} & \text{time } t_3, \end{array} \right.$$

yield $t_3 > t_2 > t_1$, suggesting indeed that the items are somehow topologically ordered in the cortex. Not all experiments however support the assumption. The idea of semantic nets is somewhat controversial (see section 10.6.3 for an application of neural networks to semantic memory by Waltz and Pollack).

2.4.5 Learning: electrophysiological studies

The observations of experimental psychology show that learning induces changes in the organization of the neuronal system, but if they give some clues regarding the nature of the changes (the Hebbian mechanism), it is not at all certain that the learning processes do work along these lines. By directly measuring the electrical activities of neurons, electrophysiology strives to elucidate where and how the changes take place. The experiments are very difficult. A few of those which we find most significant are described below.

- We have already mentioned that in the sixties Hubel and Wiesel observed that neurons of the visual cortex (area V1) of cats are selectively responsive to specific visual stimuli, to bar orientation in particular. This sensitivity builds up during a critical period in the life of the cat and the neurons of a kitten deprived of visual stimuli fail to show any bar orientation selectivity. This proved that the visual experience has indeed a deep influence on the physical properties of neurons, at least those of the visual cortex. In normal animals most neurons of the visual cortex are equally responsive to stimuli coming from either eye. If the eyelid

of one eye of a kitten is sutured, the neurons, once the suture has been removed, are only sensitive to the stimuli coming from the unsutured eye. However, after a while the neurons recover their sensitivity to both eyes.

To explain this behavior it is necessary for the synaptics dynamics of the involved neurons to be driven by the activities coming from both eyes. As one of these activities, that coming from the deprived eye, is presynaptic and the other, that coming from the other eye, is postsynaptic, the experiment has been taken as proof of the Hebbian mechanism.

- Kandel in *Aplysia* and Alkon in *Hermissenda*, two molluscs, have given further direct evidence of neuronal plasticity. Figure 2.17.a shows an experiment on *Aplysia* which is extremely close, at a neuronal level, to a classical conditioning experiment. The excitation of the neural tract A of the abdominal ganglion (a set of some hundred neurons) of *Aplysia* triggers a very small postsynaptic potential of neuron R2 (in molluscs every neuron can be identified). Excitation of tract B however makes R2 fire. The ganglion is then given simultaneous excitations from both tracts. A large enhancement of the postsynaptic potential triggered by the excitation of A, a direct proof of the Hebbian mechanism, can be observed.

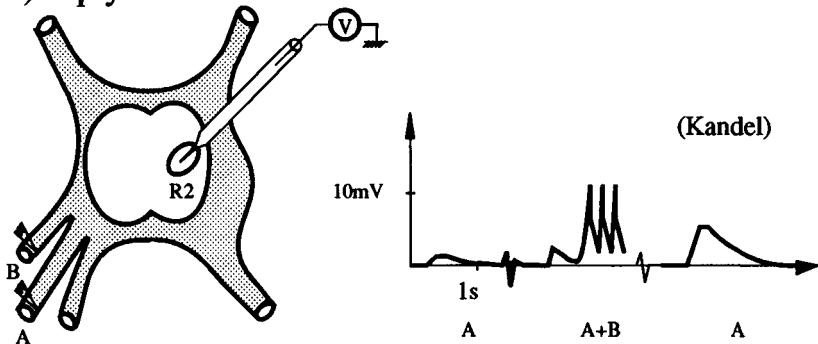
- Interesting experiments have also been carried out by Levy on the hippocampus of rabbits (see Fig. 2.17.b). The neurons are small and the experiments difficult. We have seen that the structure of the hippocampus is very regular: the connection between the neurons S1 and S2 coming from the entorhinal cortex are monosynaptic (only one synapse is involved) and homosynaptic (only binary contacts are involved). The figure shows the response of a pyramidal hippocampal neuron to a presynaptic excitation conveyed by S1. The simultaneous excitation of S1 and S2 reinforces (potentiates) that signal which fits the Hebbian paradigm. However, the excitation of the postsynaptic pathway S2, while keeping the presynaptic pathway silent, deactivates the signal. This is called an anti-Hebbian rule.

The reverse experiment, exciting S1 and keeping S2 silent, does not induce synaptic modifications. These results are gathered in Table 2.2.

		Post	
		$S_i = 0$	$S_i = 1$
Pre	$S_j = 0$	$a^{--} = 0$	$a^{-+} < 0$
	$S_j = 1$	$a^{+-} = 0$	$a^{++} > 0$

Table 2.2

a) Aplysia



b) Rat's hippocampus

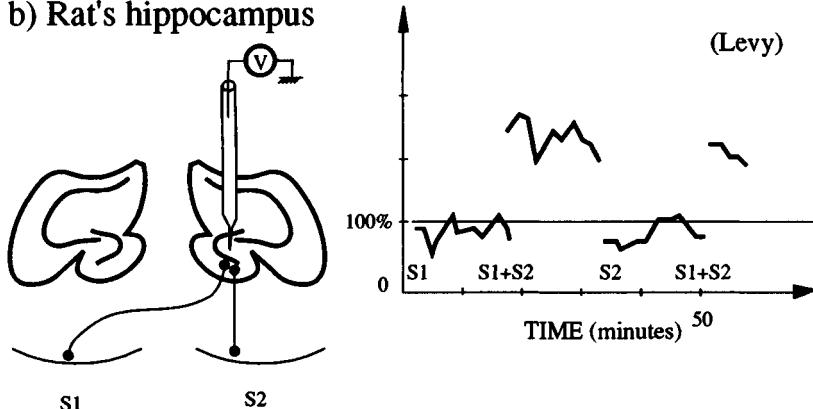


Figure 2.17. Two electrophysiological experiments which support the association learning paradigm at a neuronal level (After Kandel and Levy).

- Not all experiments are so clear-cut. The results are perturbed in particular by habituation or more involved time-dependent effects. Moreover, little is known concerning the plasticity of inhibitory synapses. Nevertheless, some general conclusions can be drawn:

- The properties of a neural network depend on a set of unary parameters, the thresholds θ_i , on a set of binary parameters, the homosynaptic efficacies J_{ij} , and on a set of ternary parameters, the hetero-synaptic efficacies J_{ijk} . All these parameters are modifiable on learning.

- The properties of a neuron as a whole can be modified. For example it has been observed that the permeability of the membrane depends on the history of the cell. This change amounts to a modification of the threshold of the neuron.

◦ There is no doubt that homo- and hetero-synapses are plastic, but they can be even more plastic than Table 2.2 suggests. Indeed it seems natural that a synapse does not undergo any modification when both afferent and efferent pathways are silent. However, the experiments have shown that activity is not compulsory for plasticity to occur. What actually matters is the membrane potential: drugs have been used to block the opening of sodium channels. When the membrane potential exceeds the threshold, the synaptic plasticity is observed in spite of the absence of spiking activity. Therefore it is all the more possible that an hyperpolarization of the membrane potential of the postsynaptic neuron, forcing it to be silent, could induce a change of the synaptic efficacy anyway. Probably the four entries of Table 2.2 are non-zero and the learning rule takes the following form:

$$\Delta J_{ij} = AS_i S_j + BS_i + CS_j + D, \quad (2.23)$$

with

$$S_i \in \{0, 1\}; \quad S_j \in \{0, 1\},$$

and

$$\left. \begin{aligned} A &= a^{++} - a^{+-} - a^{-+} + a^{--}, \\ B &= a^{+-} - a^{--}, \\ C &= a^{-+} - a^{--}, \\ D &= a^{--}. \end{aligned} \right\} \quad (2.24)$$

Models using this sort of learning rule are discussed in section 4.2.5.

◦ Finally Kandel observed, in *Aplysia*, that the heterosynaptic connections are also plastic. Let us remember that this sort of synapse is either the result of axo-axonic contacts controlling a binary synapse or that of close binary synapses. In the first instance it seems that the axo-axonic connection is non-plastic whereas the binary synapse it controls is modifiable. The axo-axonic contact would play the role of a teacher.

2.4.6 Large-scale cortical activities and the process of long-term memorization

Up to now learning has been considered as involving only local processes, but more global phenomena can also control the mechanisms of memorization.

- **Bursting.** — Generally the activity of neurons manifests itself in the form of series of spikes which show little temporal structures. Sometimes, however, neurons fire regular trains of spikes emitted at maximum frequency. These are called bursts. Bursting activities have been observed in various parts of the cortex and in the hippocampus. The important feature about this phenomenon is that a bursting activity seems to block any synaptic modification. Bursting, therefore, could be a mean of controlling memorization. The occurrence of bursting in neuronal dynamics is discussed in section 3.3.6.

- *Attention.* — According to the searchlight hypothesis of Crick, attention would be a mechanism which limits the brain activity to a specific region of the cortex. This effect would be controlled by the reticular formation with the help of sparse cortical dopaminergic neurons. One knows that attention lowers the threshold of muscular excitations. The same phenomenon could occur in the cortex: the dopaminergic receptors would lower the thresholds of neurons lying in the 'attentive' area. Memorization would be easier in this area. One has observed, for example, that a learning time in certain instrumental conditioning experiments (such as pressing a pedal to obtain food) has been reduced by a factor of two by exciting the reticular formation. In diffuse waking (an unattentive state) memorization is impossible (see section 10.5.3).

- *Dream sleep.* — Crick and Mitchinson have suggested that dream sleep would be necessary to organize the patterns memorized in the waking state. Indeed, observations have been reported showing that the heavier the memorization task in the waking state, the longer the dream sleep stage (see section 10.5.4).

Alpha waves, a synchronous activity of the cortex, develop during slow sleep. This activity apparently also blocks the memorization processes in the very same manner as bursts do. Excitation of the reticular formation desynchronizes the alpha waves.

- *The role of the hippocampus.* — The bilateral removal of the hippocampus seriously disables the long-term memorization process. However, the short-term memory mechanism seems not be affected and the items stored before the amputation are not lost. The hippocampus, therefore, must take part in the dynamics of long-term memorization but it is not the support of long-term memory itself. A mechanism which could account for the role of the hippocampus in long-term memorization is discussed in section 10.5.2. The amygdala, an organ lying at the end of the hippocampus, is also likely to be involved in memorization dynamics.

3

THE DYNAMICS OF NEURAL NETWORKS: A STOCHASTIC APPROACH

3.1 Introducing the problem

3.1.1 The formalization of neuronal dynamics

Time plays a prominent role in neuronal dynamics. The existence of a refractory period τ_r provides a standard of time to the networks. On the other hand it also takes time for the electrochemical signals to propagate along the axons, to be processed by the synapses and to diffuse at the surface of dendritic trees. These phenomena introduce delays and tend to spread the signals arriving on the hillock zone of the target neuron where the next action potential builds up. The expression (2.3) for elementary membrane potentials accounts for these effects and yields the following formula for the value $h_i(t)$ of the membrane potential of target neuron i relative to its threshold θ_i :

$$h_i(t) = \sum_j J_{ij} \int_0^\infty d\tau \chi(\tau) S_j(t - \tau_{ij} - \tau) - \theta_i. \quad (3.1)$$

Owing to the stochasticity of synaptic transmissions, $h_i(t)$ is a random variable. At time t the state of neuron i is updated according to

$$S_i(t) = \begin{cases} 1 & \text{with probability } P[S_i = 1] = P[h_i > 0], \\ 0 & \text{with probability } P[S_i = 0] = 1 - P[h_i > 0], \end{cases} \quad (3.2)$$

provided the neuron i is not in a refractory period. Otherwise the state of neuron i must remain clamped to value $S_i = 1$ at time t . $P[h_i > 0]$, the probability that the membrane potential on neuron i is positive, is given by Eq. (2.14):

$$P[h_i > 0] = \frac{1}{2} \left[1 + \operatorname{erf} \left(\frac{\bar{h}_i}{B\sqrt{2}} \right) \right] = \mathcal{S}^0(\beta \bar{h}_i), \quad (3.3)$$

with $\beta = B^{-1}$. \bar{h}_i is an average over an *ensemble* of independent albeit identical networks. S^0 is called the response function of neurons.

Remark

Most of the following developments do not depend on the precise shape of S^0 . It can be replaced by any sigmoidal curve provided that

$$\lim_{\beta \rightarrow \infty} S^0(\beta \bar{h}_i) = \mathbf{1}[\bar{h}_i], \quad \text{with } \mathbf{1}[x] = \begin{cases} 1 & \text{if } x > 0, \\ 0 & \text{if } x \leq 0. \end{cases}$$

For example, many models appeal to the following sigmoid:

$$S^0(\beta \bar{h}_i) = \frac{1}{1 + \exp(-\beta \bar{h}_i)}; \quad (3.4)$$

it obeys this asymptotic condition, is easier to compute than (3.3) and allows developments which appeal to the techniques of statistical physics (see section 3.3). By modifying β , the two curves 3.3 and 3.4 can be made very close to each other.

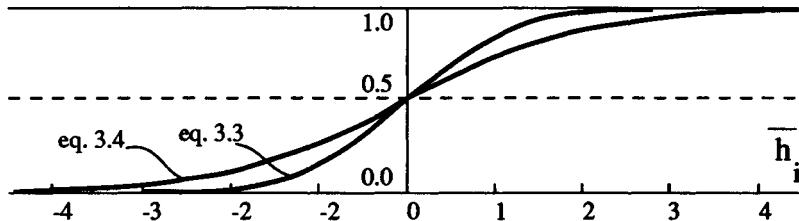


Figure 3.1. Response functions according to Eqs 3.3 and 3.4 with $\beta = 1$.

3.1.2 Early approaches to the dynamics of neural networks

The equations (3.1), (3.2) and (3.3) determine a dynamics which is very complicated. Early approaches of the neuronal dynamics *ignored the effects of the refractory period*. The average activity \bar{S}_i of a neuron i is given by

$$\begin{aligned} \bar{S}_i &= \sum_{S_i \in \{0,1\}} S_i P[S_i] \\ &= 0 \times P[S_i = 0] + 1 \times P[S_i = 1] \\ &= S^0(\beta \bar{h}_i), \end{aligned} \quad (3.5)$$

since the probability $P[S_i = 1]$ for the state of i to be $S_i = 1$ is then identical to $P[h_i > 0]$. In Eq. (2.2) we have defined an instantaneous

activity as the average $\bar{\bar{S}}_i$ of S_i over a certain time window. One says that a dynamical system is *ergodic* when time and ensemble averages are identical to one another. This point is discussed in more detail in sections 3.2.3 and 3.3.1. We shall assume that neural networks are ergodic systems and therefore that

$$\bar{S}_i \equiv \bar{\bar{S}}_i.$$

Introducing the expression (3.1) of the membrane potential into Eq. (3.5) yields:

- The Feldman and Cowan dynamical equations:

$$\overline{S_i(t)} = S^0 \left[\beta \left\{ \sum_j J_{ij} \int_0^\infty d\tau \chi(\tau) \overline{S_j(t - \tau_{ij} - \tau)} - \theta_i \right\} \right]. \quad (3.6)$$

One thus obtains a set of N integral equations which couple the average activities $\overline{S_j(t)}$ of the neurons of the network.

• The Caianiello equations. The computer is the only means of solving this complicated set of equations. For the computation to be carried out, it is necessary to have the time axis descretized. Let τ_0 be the duration of elementary time steps. Then all time variables are expressed as integer multiples of τ_0 :

- the running time, $\nu = \text{Int}(t/\tau_0)$;
- the refractory period, $\nu_r = \text{Int}(\tau_r/\tau_0)$;
- the delays, $\nu_{ij} = \text{Int}(\tau_{ij}/\tau_0)$;
- the width of the synaptic memory function χ (see section 2.3.2), $\nu_p = \text{Int}(\tau_p/\tau_0)$.

Equation (3.6) becomes

$$\overline{S_i(\nu)} = S^0 \left[\beta \left\{ \sum_j J_{ij} \sum_k \chi(k) \overline{S_j(\nu - \nu_{ij} - k)} - \theta_i \right\} \right]. \quad (3.7)$$

In the zero noise limit, $\beta \rightarrow \infty$, these equations are called the Caianiello dynamical equations:

$$S_i(\nu) = 1 \left[\sum_j J_{ij} \sum_k \chi(k) S_j(\nu - \nu_{ij} - k) - \theta_i \right], \quad (3.8)$$

with $S_i(\nu) \in \{0, 1\}$.

Although the Caianiello equations are much simpler than those of Feldman, they remain in general very complicated, at least as long as

the synaptic memory effects brought about by the function $\chi(\nu)$ are taken into account. If $\nu_p < \nu_r$ Eqs (3.8) reduce to (see also Eq. 2.4)

$$S_i(\nu) = \mathbf{1} \left[\sum_j J_{ij} S_j(\nu - \nu_{ij}) - \theta_i \right]. \quad (3.9)$$

We have seen in section 2.3.2 that ignoring the effect of synaptic memory is a legitimate approximation in many cases. However, before we focus attention on the dynamics of networks whose dynamics are determined by equations such as Eqs (3.9), it will be interesting to study a very simple case which portrays synaptic memory effects and the behavior of which has been fully elucidated by mathematicians. The analysis of this simple case proves to be a little involved. It is given in section 3.1.3. As it is not essential for the intelligibility of the rest of the text, this section can be skipped on first reading.

Remark

When symmetrical coding $\sigma_i \in \{-1, +1\}$ is used instead of binary coding $S_i \in \{0, 1\}$, the average activity $\sigma_i(t)$ is given by

$$\overline{\sigma_i(t)} = 2\overline{S_i(t)} - 1 = \mathcal{S}(\beta \overline{h}_i), \quad (3.10)$$

$$\text{with } \mathcal{S}(\beta x) = 2\mathcal{S}^0(\beta x) - 1, \quad (3.11)$$

and therefore the symmetrical response function is given by

$$\mathcal{S}(\beta x) = \operatorname{erf}(\beta x) \quad (3.12)$$

if the synaptic noise is Gaussian (Eq. 3.2), and

$$\mathcal{S}(\beta x) = \tanh(\beta x) \quad (3.13)$$

if Eq. (3.4) is used instead.

The asymptotic conditions Eq. (3.3) for \mathcal{S}^0 are then replaced by

$$\lim_{\beta \rightarrow \infty} \mathcal{S}(\beta x) = \operatorname{sign}(x) \quad (3.14)$$

The change of coding induces a renormalization of the parameters of the network through the identity $\sigma_i = 2S_i - 1$. This is made clear by looking at the mean local field \overline{h}_i ,

$$\overline{h}_i = \sum_j J_{ij} \overline{S}_j - \theta_i = \sum_j \frac{1}{2} J_{ij} \overline{\sigma}_j - \left(\theta_i - \frac{1}{2} \sum_j J_{ij} \right),$$

which we still write as

$$\bar{h}_i = \sum_j J_{ij} \bar{\sigma}_j - \theta_i,$$

where the new and old parameters correspond one another through the following relations:

$$J_{ij} \Leftrightarrow \frac{1}{2} J_{ij}, \quad \theta_i \Leftrightarrow \theta_i - \sum_j \frac{1}{2} J_{ij}. \quad (3.15)$$

Incidentally, it is often of practical interest to consider thresholding as the result of a special neuron whose state is always active, $\sigma_0 = 1$. This is made conspicuous by rewriting the membrane potential as

$$\bar{h}_i = \sum_{j=1}^N J_{ij} \bar{\sigma}_j - \theta_i = \sum_{j=0}^N J_{ij} \bar{\sigma}_j, \quad \text{with} \quad J_{i0} = -\theta_i.$$

3.1.3 A case study: a one-unit, self-inhibitory ‘neural network’

To illustrate the problems one has when postsynaptic ‘memory effects’ are introduced one considers a very simple system: a network which comprises a unique, self-inhibitory neuron ($J = -1$) and a negative threshold $-\theta$. The dynamics is noiseless, $\beta^{-1} = 0$. It is assumed that the synaptic after-effects decrease exponentially with time:

$$\chi(k) = b^k, \quad \text{with} \quad 0 < b \leq \frac{1}{2}.$$

The dynamics is determined by the following Caianiello equation:

$$S(\nu) = 1 \left[\theta - \sum_{k=0}^{K-1} b^k S(\nu - 1 - k) \right], \quad (3.16)$$

where K is the longest memory delay.

The following results were arrived at by Cosnard and Goles. Most of their demonstrations are not given here.

a) With a convenient redefinition of the threshold, the dynamics determined by Eq. (3.16) is identical to the dynamics determined by

$$S(\nu) = 1 \left[\theta^* - \sum_{k=0}^{K-1} 2^{-k} S(\nu - 1 - k) \right], \quad (3.17)$$

where the new threshold θ^* is given by

$$\theta^* = \sum_{k=0}^{K-1} 2^{-k} S_k^*. \quad (3.18)$$

In Eq. (3.18) the S_k^* 's are the set of neuronal states which minimizes the quantity:

$$\left| \theta - \sum_{k=0}^{K-1} b^k S_k^* \right|. \quad (3.19)$$

Proof. — Let $\{S_k \equiv S(\nu - 1 - k)\}$, $k = 0, \dots, K - 1$, be a string of states generated by the dynamics (3.16). The state $S(\nu)$ of the neuron at time ν is determined by the sign of

$$\theta - \sum_k b^k S_k.$$

The sign of this expression is the same as that of

$$\theta - \sum_k b^k S_k - \left[\theta - \sum_k b^k S_k^* \right] = \sum_k b^k (S_k^* - S_k),$$

owing to the definition (3.19).

Since $b < \frac{1}{2}$, one has $b^k > \sum_{\ell > k} b^\ell$ and the sign of the sum, that is the state $S(\nu)$, is determined by the first non-zero term $S_k^* - S_k$.

On the other hand, by using Eq. (3.18) one obtains

$$\theta^* - \sum_k 2^{-k} S_k = \sum_k 2^{-k} (S_k^* - S_k).$$

The first non-zero term is the same in this sum as in the previous one. Therefore the signs of both sums are the same and the string of states determined by Eq. 3.16 is identical to that determined by Eq. 3.17. The conclusion is that it is enough to study the dynamics of the system using $b = \frac{1}{2}$. The behavior of the system depends on a unique variable, the threshold θ .

- b) If $\theta < 0$ then $S(\nu) = 0$ for all ν s;
- if $\theta > \sum_{k=0}^{K-1} b^k$ then $S(\nu) = 1$ for all ν s;
- if $\theta < 1$ the dynamics cannot generate two successive 1s.

Proof. — Let us assume that $S(\nu) = S(\nu + 1) = 1$; then

$$\begin{aligned} S(\nu + 1) &= 1 \left[\theta - \sum_{k=0}^{K-1} 2^{-k} S(\nu - k) \right] \\ &= 1 \left[\theta - S(\nu) - \sum_{k=1}^{K-1} 2^{-k} S(\nu - k) \right] \\ &= 1 \left[\theta - 1 - \sum_{k=1}^{K-1} 2^{-k} S(\nu - k) \right] = 1, \end{aligned}$$

which implies $\theta > 1$.

We now give a series of results without proof.

- c) The dynamics determined by Eq. (3.16) admits a unique limit cycle, a non-trivial result. Obviously, the length T of the cycle is $T \leq K + 1$.

Let us choose a length T , ($T \leq K + 1$) and C , ($C \leq T$) a number of 1s in the cycle. Then:

- There always exists a threshold θ which triggers limit cycles of length T .
- C/T is an irreducible fraction and $C/T \leq \frac{1}{2}$ for $\theta < 1$ (a consequence of the second theorem).
- Let us choose an irreducible fraction C/T . Then there always exists a threshold such as the dynamics admits a cycle with C/T as its irreducible fraction.
- C/T is a non-decreasing function of θ (an intuitive result).

We give an example with $K = 8$. All cycles with $T = 2, 3, \dots, 9$ do exist. There are 14 irreducible fractions fulfilling the above conditions:

T	2	3	4	5	6	7	8	9
C/T	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{5}, \frac{2}{5}$	$\frac{1}{6}$	$\frac{1}{7}, \frac{2}{7}, \frac{3}{7}$	$\frac{1}{8}, \frac{3}{8}$	$\frac{1}{9}, \frac{2}{9}, \frac{4}{9}$

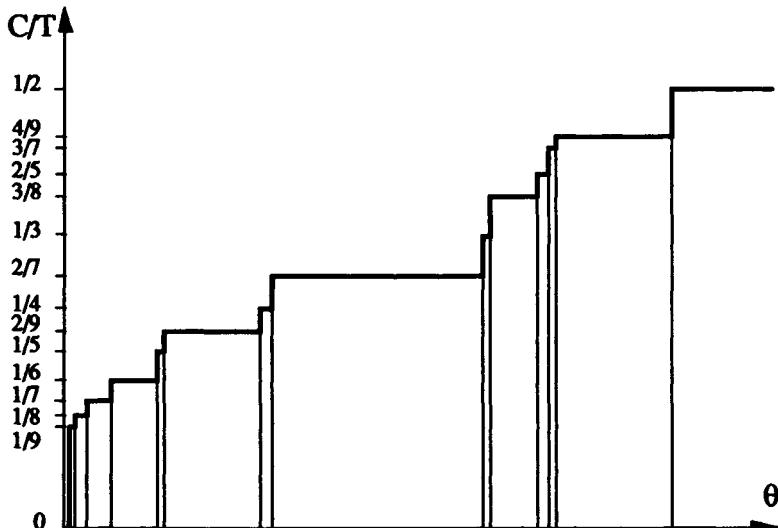


Figure 3.2. The dynamics of a one-unit, self-inhibitory neuron: a 'devil staircase' (After Cosnard and Goles).

Let us consider the whole set of partial sums of b^k . Obviously the nature of the dynamics depends only on the position of the threshold with respect to these sums. Rearranging the fractions C/T in increasing order, one finds the results displayed in Fig. 3.2 and in Table 3.1.

C/T	T	Cycle	Threshold ($b = \frac{1}{2}$)
0	1	0 0 0 0 0 0 0 0 0	
$\frac{1}{9}$	9	100000000 100000000	$b^7 = 0.0078$
$\frac{1}{8}$	8	10000000 10000000	$b^6 = 0.0156$
$\frac{1}{7}$	7	1000000 1000000	$b^5 = 0.0312$
$\frac{1}{6}$	6	100000 100000 100000	$b^4 = 0.0625$
$\frac{1}{5}$	5	10000 10000 10000	$b^3 = 0.1250$
$\frac{2}{9}$	9	100001000 100001000	$b^3 + b^7 = 0.1328$
$\frac{1}{4}$	4	1000 1000 1000 1000	$b^2 + b^6 = 0.2656$
$\frac{2}{7}$	7	1000100 1000100	$b^2 + b^5 = 0.2812$
$\frac{1}{3}$	3	100 100 100 100 100	$b + b^4 + b^7 = 0.5703$
$\frac{3}{8}$	8	10010010 10010010	$b + b^4 + b^6 = 0.5781$
$\frac{2}{5}$	5	10010 10010 10010	$b + b^3 + b^6 = 0.6406$
$\frac{3}{7}$	7	1001010 1001010	$b + b^3 + b^5 = 0.6562$
$\frac{4}{9}$	9	100101010 100101010	$b + b^3 + b^5 + b^7 = 0.6640$
$\frac{1}{2}$	2	10 10 10 10 10 10 10	$1 + b^2 + b^4 + b^6 = 0.8281$
1	1	1 1 1 1 1 1 1 1 1	

Table 3.1

C/T is a monotone increasing function with zero derivative almost everywhere. This is a discrete approximation of a ‘devil staircase’. It shows how sensitive the characteristics of limit cycles are to slight variations of the threshold.

3.2 Noiseless neural networks

3.2.1 The neural dynamics can be cast in the form of a set of first-order equations

In this section we keep on ignoring the blocking effect due to the refractory period and we consider that the assumption of short enough postsynaptic potentials is verified: there are no synaptic memory effects. We also assume that the noise level is zero, $\beta^{-1} = 0$. Noise is introduced in section 3.3. In the framework defined by these hypotheses the dynamics of a neural network Σ , defined by its set of parameters $\{J_{ij}\}$, $\{\nu_{ij}\}$ and θ_i , is given by a Caianiello equation (cf. Eq. (3.9)):

$$S_i(\nu) = 1 \left[\sum_j J_{ij} S_j(\nu - \nu_{ij}) - \theta_i \right].$$

This is a deterministic dynamics but delays make it a high-order dynamics. Indeed to start the computation of $S_i(\nu = 0)$ it is necessary to

know the values of S_j at times $-1, -2, \dots, -\nu_{ij}$.[†] We shall see that standard models of neural networks are built on first-order dynamics. All delays are then equal ($\nu_{ij} = 1$), and the state $I(\nu)$ of the system at time ν is fully determined by the state $I(\nu - 1)$ of the system at time $(\nu - 1)$. There are good reasons, which we explain below, for accepting this simplification in small enough networks, but considering the effect of delays is unavoidable for large systems. This is why we first consider dynamics involving delays.

As a matter of fact it is always possible to transform sets of high-order equations into larger sets of first-order equations. The transformation helps in programming the equations on computers. It also helps in analyzing the behavior of the network. The price to pay is the introduction of extra dynamical variables. The space spanned by the dynamical variables is called the *phase space* of system Σ .

We therefore introduce dynamical variables $S_j^\kappa(\nu)$ defined by

$$S_j^\kappa(\nu) = S_j(\nu - \kappa), \quad k = 0, 1, \dots \quad (3.20)$$

In particular, $S_i^0(\nu) = S_i(\nu)$. Then, using the two following identities

$$S_j(\nu - \nu_{ij}) \equiv S_j((\nu - 1) - (\nu_{ij} - 1)) \equiv S_j^{\nu_{ij}-1}(\nu - 1)$$

and $S_j^\kappa(\nu) \equiv S_j((\nu - 1) - (\kappa - 1)) \equiv S_j^{\kappa-1}(\nu - 1)$,

the set of high-order equations Eqs (3.9) is replaced by the following set of first-order equations:

$$\left. \begin{aligned} S_i^0(\nu) &\equiv S_i(\nu) = \mathbf{1} \left[\sum_j J_{ij} S_j^{\nu_{ij}-1}(\nu - 1) - \theta_i \right], \\ &\quad \text{for } i = 1, \dots, N, \\ S_j^\kappa(\nu) &= S_j^{\kappa-1}(\nu - 1) \quad \text{for } \kappa = 1, 2, 3, \dots, \nu_{ij} - 1, \end{aligned} \right\} \quad (3.21)$$

with the following initial conditions:

$$S_j^{\kappa-1}(-1) = S_j(-\kappa) \quad \text{for } \kappa = 1, 2, \dots, \nu_j^*.$$

ν_j^* , the number of variables S_j^κ , is given by

$$\nu_j^* = \max_i(\nu_{ij}). \quad (3.22)$$

The total number N^* of variables S_i^κ is

$$N^* = \sum_j \nu_j^*.$$

[†] See a few comments on orders of dynamics at the end of this section.

N^* reduces to N the number of neurons, when all delays are equal:

$$\nu_{ij} = 1, \quad \forall i, j \quad (\text{then } \nu_j^* = 1).$$

On orders of differential equations

A differential equation such as $\frac{dy}{dt} = f(y)$ is first-order since it can be transformed into the discrete equation (taking $\tau_0 = 1$) $y(\nu) - y(\nu - 1) = f[y(\nu - 1)]$ or

$$y(\nu) = g[y(\nu - 1)], \quad \text{with } g = 1 + f,$$

and the knowledge of $y(-1)$ is enough to determine the whole trajectory.

Similarly it is necessary to know $y(-1)$ and $y(-2)$ to find a solution of the second-order equation

$$\frac{d^2y}{dt^2} = f\left(y, \frac{dy}{dt}\right),$$

once it has been written in the form of a discrete equation:

$$y(\nu) = 2y(\nu - 1) - y(\nu - 2) + f\left[y(\nu - 1), (y(\nu - 1) - y(\nu - 2))\right].$$

It is easy to transform this second-order equation into a set of two first-order equations:

$$\begin{aligned} z(\nu) &= z(\nu - 1) + f[y(\nu - 1), z(\nu - 1)], \\ y(\nu) &= y(\nu - 1) + z(\nu). \end{aligned}$$

Let now consider the seemingly simple equation

$$\frac{dy}{dt} = f[y(t - \tau)].$$

The order of this equation is not one, but it is rather infinite because it is necessary to know $y(t)$ in the whole range $t \in [0, -\tau]$ to have it solved. The discretized version of the equation is

$$y(\nu) = y(\nu - 1) + f[y(\nu - \nu_0)], \quad \text{with } \nu_0 = \tau/\tau_0.$$

The initial conditions are a set of values

$$\{y(\nu)\} \quad \text{for } \nu = -1, -2, \dots, -\nu_0.$$

One knows that systems driven by more than two non-linear first-order equations are bound to show chaotic behaviors. Delays increase the number of dynamical variables, and therefore they tend to make the networks more chaotic. For example, a system obeying these very simple dynamics,

$$\frac{dy}{dt} = \alpha \sin[\omega(t - \tau)],$$

shows a whole set of complex behaviors.

3.2.2 General properties of first-order deterministic dynamics

A state I of the network is a collection of N^* bits:

$$I(\nu) = \left\{ S_i^\kappa(\nu) ; i = 1, \dots, N ; \kappa = 0, 1, \dots, \nu_i^* - 1 \right\}.$$

That is, $I \in \{0, 1\}^{N^*}$ and the number of possible states I is

$$\mathcal{N} = 2^{N^*}.$$

A given state I can be viewed as a vertex of a hypercube embedded in the N^* -dimensional phase space of the system. The dynamics makes the point which represents the state of the network wander from one vertex to another. The equations of motion (3.21) associate a successor state J to every state I . All information regarding the dynamics can therefore be trapped into a *transition matrix* \mathbf{W} whose elements $W(I | J)$ are given by

$$W(I | J) = \begin{cases} 1 & \text{if } I \text{ is the successor of } J, \\ 0 & \text{otherwise.} \end{cases}$$

\mathbf{W} is a $2^{N^*} \times 2^{N^*}$ matrix. It is to be operated upon a 2^{N^*} column state vector $\tilde{\rho}(\nu)$ which determines the state of Σ . The entries $\rho(J, \nu)$ of $\tilde{\rho}$ are given by

$$\rho(J, \nu) = \begin{cases} 1 & \text{if the system is in state } J \text{ at time } \nu, \\ 0 & \text{otherwise.} \end{cases}$$

The elements of $\tilde{\rho}$ are all zero except one entry which is 1.

The equation of the dynamics is given by

$$\rho(I, \nu) = \sum_J W(I | J) \rho(J, \nu - 1). \quad (3.23)$$

Equation (3.23) is called the *master equation*.

The simple structure of the matrix \mathbf{W} allows the possible behaviors of the network to be easily classified. \mathbf{W} is made of columns comprising one and only one non-zero element. Then the states can be relabeled in such a way that the matrix displays a block structure (see Fig. 3.3). The set of states which labels a block makes a *basin of attraction*. Owing to the deterministic character of the dynamics each block comprises a unique cyclic submatrix. The set of states connected by this submatrix is the *(cyclic) attractor* of the basin. The length of the cycle is the size of the submatrix. When the length is 1 the cycle is a *fixed point* into which all trajectories of the basin flow. The maximum length of a cycle is 2^{N^*} (the Poincaré period). When the cycle is of the order of 2^{N^*} long the behavior

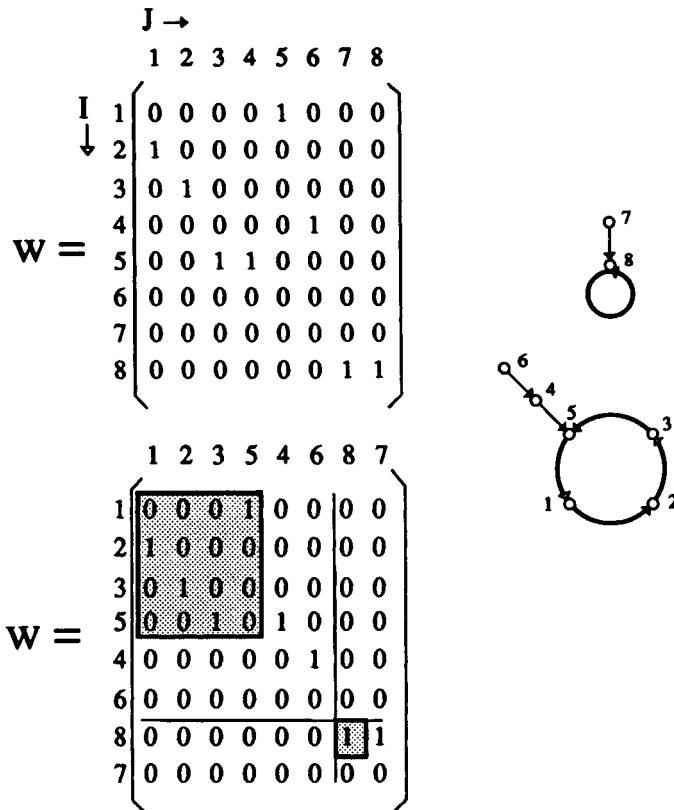


Figure 3.3. An example of a deterministic dynamics. The evolution matrix W involves 8 states and 2 basins of attraction. The attractor of one basin is a fixed point, state 8. The attractor of the other basin is a 4 state cycle made of states 5, 1, 2 and 3. States 6 and 7 are boundary states. State 4 is a transient state.

is chaotic. The states not belonging to the attractor are either boundary states or transient states. A state I such that $W(I | J) = 0$ for all J s is a boundary state.

3.2.3 Standard neuronal dynamics

a) *On ensemble averages.* — The history of a given network depends theoretically on the set $\{\nu_{ij}\}$ of its delays. However, macroscopic tasks, such as pattern recognition and effector coordination and even higher tasks such as problem solving, involve the recruitment of tens or hun-

dreds of thousands of neurons, and times needed to carry out the tasks are of the order of 100 ms. It is therefore unlikely that the performances of the network depend on the precise distribution of delays. This remark prompts us to think that as far as one is interested by the computation of typical properties, two networks which have the same given set of efficacies $\{J_{ij}\}$ but two different sets of delays $\{\nu_{ij}\}$ are equivalent, provided that the sets of delays display some similarities. For example, they must have (at least) the same average value $\bar{\nu}$ and the same variance. A set of networks which fulfills these conditions forms an *ensemble* and the typical properties of the network may be computed by carrying out averages on the ensemble.

The very same problem arises in the theory of large physical systems. The purpose of statistical physics is the computation of macroscopic properties given some general constraints \mathcal{C} that a system Σ has to obey. The measurement of a property O carried out on a set of N^s systems Σ^s equivalent to Σ must yield about the same results provided they are subject to the same constraints \mathcal{C} . Therefore the physical quantities of interest are statistical averages \bar{O} performed on the set. The average is given by

$$\overline{O(\nu)} = \frac{1}{N^s} \sum_I O(I) N^s(I, \nu), \quad (3.24)$$

where $N^s(I, \nu)$ is the number of systems Σ^s which are in state I at time step ν . The probability distribution of states at time ν , $\rho(I, \nu)$, is the relative number of states which are in state I at time ν :

$$\rho(I, \nu) = \frac{N^s(I, \nu)}{N^s}.$$

$\rho(I, \nu)$ obeys the normalization condition

$$\sum_I \rho(I, \nu) = 1, \quad (3.25)$$

and the statistical average is therefore given by

$$\overline{O(\nu)} = \sum_I O(I) \rho(I, \nu). \quad (3.26)$$

It is convenient to define a distribution vector $\tilde{\rho}(\nu) = \{\rho(I, \nu)\}$. Since $\tilde{\rho}$ is a linear combination of vectors which all obey a master equation, its dynamics are also determined by a master equation (3.23):

$$\tilde{\rho}(\nu) = \mathbf{W} \cdot \tilde{\rho}(\nu - 1),$$

where $W(I | J)$ is the probability for a system to be in state I at time ν given that it is in state J at time $(\nu - 1)$ and

$$\sum_I W(I | J) = 1,$$

a condition which defines the so-called statistical matrices.

Ensembles and statistical mechanics

The properties of an ensemble of equivalent systems depend heavily on what the systems have in common, that is on the nature of constraints C . The constraints can deal with the initial conditions of systems Σ^* : any initial condition is permitted provided they fulfill some general properties C .

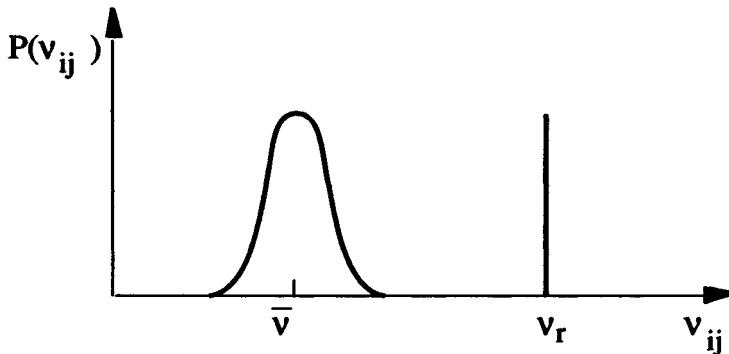
Let us consider a simple experiment, namely a Joule Thomson experiment. A system is made of two compartments; one is filled with a gas at temperature T and pressure P , the other is empty. At time $\nu = 0$ the valve connecting the compartments is opened and one follows how the temperature $T(\nu)$ and the pressure $P(\nu)$ evolve with time ν . In this experiment the initial state I , $\rho(J, \nu = 0)$, is the set of positions and velocities of all molecules of the gas at time $\nu = 0$. This state depends on the moment the valve is opened. Nevertheless, the experiment shows that the results do not depend much on this parameter. This must be related to the fact that all initial states corresponding to different opening times obey the same set of constraints C : the geometry of the different experiments and the nature of the molecules are the same, the vessel contains the same number N of molecules, the molecules and their momenta are uniformly distributed in the vessel. Moreover, the average velocities of the molecules are \sqrt{T} for all initial conditions. Experiments carried out with different vessels s ; $s = 1, 2, \dots, N^*$, all satisfying the same set of constraints C , also yield the same macroscopic results, which justifies the idea of ensemble averages.

Constraints on initial conditions are not the sole possibility, however. One can imagine that some of the parameters which determine a system are not that rigid. For example, the walls of the different vessels are certainly not identical. The fluctuations of surfaces change the trajectories of the molecules of the gas even though the initial conditions are exactly the same. Then sample averages can be computed on a set of vessels whose surface conditions can vary to a point that satisfies some general constraints C . The same idea is put at work in neural networks by considering ensembles of nets with various sets of delays obeying some global constraint C .

The effects of the refractory period have been ignored so far. They are now reintroduced. For non-myelinated axons, the distance which information covers in one refractory period is $500 \text{ cms}^{-1} \times 0.005 \text{ s} = 2.5 \text{ cm}$ (close neurons are related through non-myelinated neurons). Therefore, in networks of that size, the average delay $\bar{\nu}$ and the refractory period ν_r have the same magnitude. $\bar{\nu}$ and ν_r determine two time scales for neural networks and according to whether the average delay is smaller, of the order of, or larger than the refractory period one distinguishes three types of neuronal dynamics.

b) *The Glauber dynamics.* — We first examine the situation with $\nu_r > \bar{\nu}$, which is the case for small enough systems, typically smaller than 1 cm.

The interplay between the role of the refractory period and that of a distribution of delays may be approximately described by choosing a neuron at random and by updating its state according to its local field. This is the Glauber (Monte-Carlo) dynamics



To give a few justifications to the algorithm we consider a member Σ of an ensemble of equivalent systems Σ^* . The state S_i of a neuron i of Σ is blocked as long as i is in a refractory period and the moment it can be updated does not depend on the state of the network at that moment. As it is unlikely that two neurons are updated at the very same time, one can choose the standard of time τ_0 in such a way that only one neuron is updated at every time step. If one now considers a whole ensemble of systems Σ^* , there is no clue regarding the label of the neuron to be updated. Therefore the dynamics amounts to choosing a neuron at random at every time step, to compute its local field and to update its state accordingly.

- [1) At time ν choose a neuron i *at random*.
- 2) Compute its local field h_i :
$$h_i(\nu) = \sum_j J_{ij} S_j(\nu - 1) - \theta_i.$$
- 3) Update the state S_i according to:
$$S_i(\nu) = \mathbf{1}[h_i(\nu)].$$
- 4) Iterate in 1).

The Glauber dynamics for noiseless networks.

Remarks on the Glauber dynamics

- The algorithm accounts for the effect of the refractory period only statistically: no mechanism hinders the updating of the very same neuron

in two successive time steps.

- The dynamics are first-order, serial and asynchronous.
- Because N neurons at most can be updated in one refractory period, the standard time is given by

$$\tau_0 = \frac{\tau_r}{N}. \quad (3.27)$$

This provides a natural standard of time for the duration of Monte-Carlo steps in computer simulations of neural networks .

- The role of delays has been dismissed on the account that they are randomly generated during the epigenesis of the network and frozen afterwards. This hypothesis could not be justified if learning selects among all possible circuits, resonant paths which make several action potentials arrive synchronously on certain neurons. It must be realized however that for networks whose dimensions stretch over a few millimeters, the axons, when they are active, fire on their entire lengths and synchronization is not likely. The situation can be very different with larger networks.

- Since the dynamics is first-order $N^* = N$ and a state of the network is $I = \{S_i\} \in \{0,1\}^N$. The phase space can be imagined as an N -dimensional hypercube whose vertices represent the 2^N states I of the network. Because the Glauber dynamics allow the flipping of one neuronal state at a time, the neighborhood of a vertex J on the cube, that is the set of vertices I one can reach in one time step, is limited to the N nearest neighbors of J . This determines the non-diagonal elements of the transition matrix \mathbf{W} (see Eq. (3.48) below):

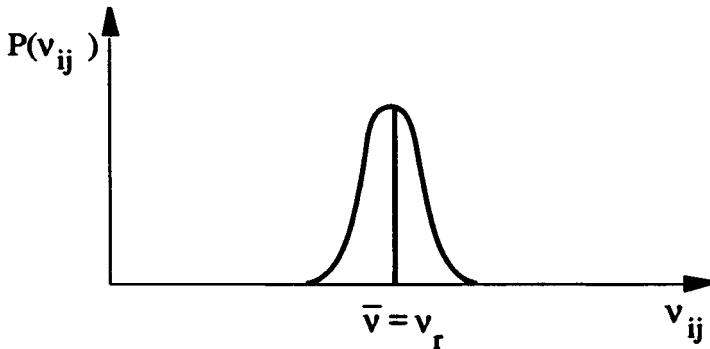
$$W(I | J) = \begin{cases} \frac{1}{N-1} \mathbf{1}[\sigma_i(I)h_i(J)] & \text{if } I \text{ is a neighbor} \\ & \text{of } J \text{ and } I \neq J, \\ 0 & \text{if } I \text{ is not a neighbor of } J. \end{cases} \quad (3.28)$$

Since \mathbf{W} is a statistical matrix, one has

$$W(J | J) = 1 - \sum_{I \neq J} W(I | J). \quad (3.29)$$

\mathbf{W} is a very sparse matrix. The total number of its elements is $(2^N)^2 = 2^{2N}$ and the number of non-zero elements is only $2^N \times N$.

- c) *The Little dynamics.* — If $\nu_r \simeq \bar{\nu}$ the signals do not interfere on their way from one neuron to the other.



The dynamics is parallel (synchronous). An approximate model consists in choosing

$$\tau_0 = \tau_r. \quad (3.30)$$

The updating of all neurons is carried out simultaneously and the corresponding algorithm is called the Little dynamics.

- 1) Compute *all* local fields

$$h_i(\nu) = \sum_j J_{ij} S_j(\nu - 1) - \theta_i.$$
- 2) Update all neurons according to

$$S_i(\nu) = \mathbf{1}[h_i(\nu)].$$
- 3) Iterate in 1).

The Little dynamics for noiseless networks.

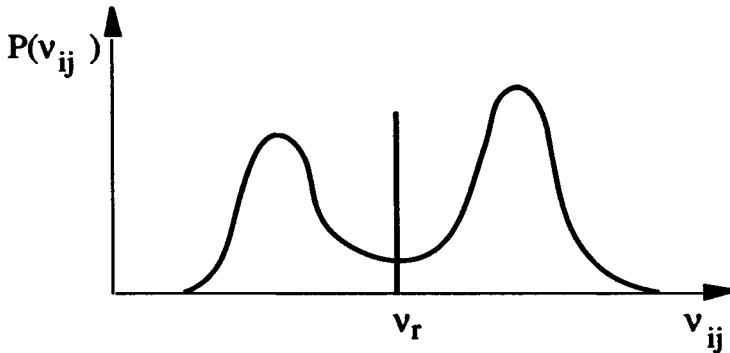
The non-diagonal elements of \mathbf{W} are given by (see Eq. (3.46))

$$\mathbf{W}(I \mid J) = \prod_{i=1}^N \mathbf{1}[\sigma_i(I) h_i(J)].$$

Remarks on the Little dynamics

- The dynamics is first-order, parallel and synchronous.
- The transition matrix \mathbf{W} of the Little dynamics is that of a first-order deterministic dynamics with only one non-zero element in every one of its 2^N columns.

d) *Large systems.* — Finally, there remains the case where $v_r < \bar{v}$ or more complicated situations such as the following:



It is then necessary to appeal to equations such as (3.21) which involve the explicit introduction of delays.

We have seen that biological systems, the cortex for example, comprise structures of different sizes, some being small enough for the Glauber algorithm to be applicable whereas at a higher level the range of interactions is too large for the delay effects to be ignored. The dynamics of such complicated structures could be decoupled into two processes: on the one hand structures of a size similar to or smaller than that of columns would follow fast Glauber dynamics, and on the other a slow, delay-dependent dynamics would drive the interactions between columns. According to this picture, the relaxation of a cortical column puts the neurons of that column only into play. It is the information contained in one of its stationary states that a column sends to the columns it is linked with. These interactions are delay-dependent and intercolumnar dynamics could eventually be observed through evoked potentials. If the delays between columns are of the same order, the columnar activities can again be modeled by a Little dynamics, using the stationary states of columns as dynamical variables instead of neuronal states themselves.

3.2.4 Parallel dynamics and Lyapunov function

In this section we give some results regarding the nature of asymptotic behaviors of neural networks whose dynamics is of Little type, that is, first-order and parallel. In forthcoming developments the symmetrical coding $\sigma_i \in \{+1, -1\}$ is adopted.

Assuming that the thresholds vanish, $\theta_i = 0$, the Little dynamics of noiseless neural networks is driven by the following rules:

$$\sigma_i(\nu + 1) = \begin{cases} +1 & \text{if } \sum_{j=1}^N J_{ij} \sigma_j(\nu) > 0, \\ -1 & \text{if } \sum_{j=1}^N J_{ij} \sigma_j(\nu) < 0. \end{cases}$$

The inequalities can be combined, leading to

$$\sigma_i(\nu + 1) \sum_j J_{ij} \sigma_j(\nu) > 0, \quad (3.31)$$

and it is natural to define a *two-time Lyapunov function* as

$$L(\nu, \nu + 1) = - \sum_i \sum_j J_{ij} \sigma_j(\nu) \sigma_i(\nu + 1). \quad (3.32)$$

Let us consider the change of the Lyapunov function in one time step:

$$\begin{aligned} \Delta L &= L(\nu + 1, \nu + 2) - L(\nu, \nu + 1) \\ &= \sum_i \sum_j \left[J_{ij} \sigma_j(\nu + 1) \sigma_i(\nu + 2) - J_{ij} \sigma_j(\nu) \sigma_i(\nu + 1) \right]. \end{aligned}$$

Swapping i for j in the second term, this becomes:

$$\Delta L = - \sum_i \left\{ \sigma_i(\nu + 2) \left[\sum_j J_{ij} \sigma_j(\nu + 1) \right] - \sigma_i(\nu) \left[\sum_j J_{ji} \sigma_j(\nu + 1) \right] \right\}.$$

When the interactions J_{ij} are given no special symmetry, there is nothing to say about ΔL . We therefore introduce constraints on efficacies. In actual fact we consider two types of networks:

$$J_{ji} = k J_{ij},$$

with $k = 1$ for symmetrically connected networks and $k = -1$ for antisymmetrically connected networks.

Letting $\sigma_i(\nu) = \kappa^{(i)} \sigma_i(\nu + 2)$ with

$$\kappa^{(i)} = \begin{cases} 1 & \text{if } \sigma_i(\nu + 2) = \sigma_i(\nu), \\ -1 & \text{if } \sigma_i(\nu + 2) = -\sigma_i(\nu), \end{cases}$$

ΔL is given by

$$\begin{aligned} \Delta L &= - \sum_i (1 - k \kappa^{(i)}) \left[\sigma_i(\nu + 2) \sum_j J_{ij} \sigma_j(\nu + 1) \right] \\ &= \sum_i (k \kappa^{(i)} - 1) x_i, \end{aligned}$$

with $x_i = \sigma_i(\nu + 2) \sum_j J_{ij} \sigma_j(\nu + 1) > 0$ according to Eq. (3.31). L is a monotonous function if

$$\text{either } k \kappa^{(i)} < 1 \quad \text{or} \quad k \kappa^{(i)} > 1 \quad \forall i.$$

a) If $k = 1$, the network is symmetrically connected: $J_{ij} = J_{ji}$. Then

$$\Delta L = \sum_i (\kappa^{(i)} - 1) x_i$$

and $\Delta L \leq 0$ since $\kappa^{(i)} = \pm 1$: the Lyapunov function is monotonous and non-increasing and since it is a bounded function it must stop somewhere. When it is stationary one has:

$$\Delta L = 0$$

and therefore $\kappa^{(i)} = 1$, $\forall i$, which implies that $\sigma_i(\nu + 2) = \sigma_i(\nu)$. That means that either $\sigma_i(\nu + 1) = -\sigma_i(\nu)$ for all or part of neurons i and the limit behaviors are cycles of length 2, or $\sigma_i(\nu + 1) = \sigma_i(\nu)$ for all neurons and the limit behaviors are fixed points.

One concludes that the asymptotic behavior of the parallel dynamics of neural networks with symmetrical interactions and zero thresholds are *either fixed points or limit cycles of length two*.

b) If $k = -1$ then $J_{ij} = -J_{ji}$ and the interactions are antisymmetrical. One has

$$\Delta L = \sum_i (\kappa^{(i)} + 1)x_i.$$

The Lyapunov function still decays. It is stationary when

$$\kappa^{(i)} = -1, \quad \forall i.$$

Then $\sigma_i(\nu + 2) = -\sigma_i(\nu)$, that is $\sigma_i(\nu + 4) = \sigma_i(\nu)$, which means that the asymptotic behaviors of zero-threshold, antisymmetrically connected networks are necessarily *limit cycles of length 4*. These demonstrations which have been given by Goles do not hold, however, when the thresholds θ_i are non-zero. In this case one must appeal to another monotonous function which is called the energy function.

3.2.5 Asynchronous dynamics and the energy function

The *energy function* is defined as

$$H(\nu) = -\left[\frac{1}{2} \sum_i \sum_j J_{ij} \sigma_i(\nu) \sigma_j(\nu) - \sum_i \theta_i \sigma_i(\nu) \right]. \quad (3.33)$$

This is a one-time Lyapunov function. It depends only on the current state, not on two states as in the former function. Therefore a scalar value $H(I)$ can be attached to every corner of the N -dimensional hypercube which represents the phase space of the network.

There is nothing special to say about the variation

$$\Delta H = H(\nu + 1) - H(\nu)$$

undergone by the energy function in one time step, except when:

- a) the dynamics is asynchronous, with one unit being allowed to change its state at every time step (the Glauber dynamics);
- b) the self-connections vanish, $J_{ii} = 0$: then the double summation entering the definition of H can be looked upon as a summation over the (non-oriented) links between the units:

$$\frac{1}{2} \sum_i \sum_j \cdots \mapsto \sum_{\langle ij \rangle} \cdots.$$

Let i be the neuron which is at stake at time ν . Then

$$\begin{aligned}\Delta H = \Delta H_i = & -\left[\frac{1}{2} \sum_j (J_{ij} + J_{ji}) \sigma_j(\nu + 1) - \theta_i\right] \sigma_i(\nu + 1) + \cdots \\ & + \left[\frac{1}{2} \sum_j (J_{ij} + J_{ji}) \sigma_j(\nu) - \theta_i\right] \sigma_i(\nu)\end{aligned}$$

and

$$\Delta H_i = -\left[\frac{1}{2} \sum_{j \neq i} (J_{ij} + J_{ji}) \sigma_j(\nu) - \theta_i\right] (\sigma_i(\nu + 1) - \sigma_i(\nu)),$$

since $\sigma_j(\nu + 1) = \sigma_j(\nu)$ for $j \neq i$.

On the other hand, by letting

$$\sigma_i(\nu + 1) - \sigma_i(\nu) = \kappa^{(i)} \sigma_i(\nu + 1)$$

$$\text{with } \kappa^{(i)} = \begin{cases} 0 & \text{if } \sigma_i(\nu + 1) = +\sigma_i(\nu), \\ 2 & \text{if } \sigma_i(\nu + 1) = -\sigma_i(\nu), \end{cases}$$

one has

$$\Delta H_i = -\kappa^{(i)} \sigma_i(\nu + 1) \left[\frac{1}{2} \sum_j (J_{ij} + J_{ji}) \sigma_j(\nu) - \theta_i \right],$$

which yields a set of relations which can be compatible only when the network is symmetrically connected, $J_{ij} = J_{ji}$. H is of no use for antisymmetrically connected networks (since, if the case arises, $\Delta H \equiv 0$). Then

$$\Delta H_i = -\kappa^{(i)} \sigma_i(\nu + 1) \left[\sum_{j \neq i} J_{ij} \sigma_j(\nu) - \theta_i \right]$$

and the fundamental equation of the dynamics,

$$\left[\sum_j J_{ik} \sigma_j(\nu) - \theta_i \right] \sigma_i(\nu + 1) \geq 0,$$

compels the energy to be a non-increasing function of time since $\kappa^{(i)}$ is non-negative.

When the energy reaches its lower bound, that is when H is stationary, $\kappa^{(i)} = 0$ and

$$\sigma_i(\nu + 1) = \sigma_i(\nu), \quad \forall i.$$

All neuronal states are frozen. The conclusion is that the asymptotic behaviors of the asynchronous dynamics of symmetrically connected

networks are *fixed points*. The thresholds can take non-zero values but *the neurons must not be self-connected*. The above derivation proved that energy defined as a quadratic function of neuronal states is of relevance to symmetrically connected networks only, a reminder of Newton's law of action and reaction. However, if the symmetry condition is a sufficient condition to ensure that the limit behaviors of the dynamics are fixed points, it is by no means a necessary condition. We shall see that asymmetrically connected networks can induce this type of asymptotic behavior as well.

On attractors of symmetrical networks

We have seen that the attractors of symmetrical networks are either cycles of length 2 or cycles of length 1 (fixed points) when the dynamics are parallel and cycles of length 1 when the dynamics are serial. This last result has been obtained without making any reference to the order of the units to be updated and therefore it remains valid if the neurons are updated along a well-defined order.

Many other sorts of dynamics can be envisioned. For example, block dynamics consists in updating a block of N^b neurons at every time step, with $N^b = N$ in parallel dynamics and $N^b = 1$ in serial dynamics. The neurons of a block are chosen at random in stochastic dynamics and the series of blocks is well defined in deterministic dynamics. Deterministic *symmetrical* networks can display attractors of any sort, limit cycles of any length, provided the symmetrical interactions and the series of blocks are conveniently chosen. Figure 3.4 shows an example of a 4-unit network fully connected through excitatory interactions with a series of blocks which determines a limit cycle of length 3.

3.3 Taking synaptic noise into account

3.3.1 *The neural systems as networks of probabilistic threshold automata*

Up to now noise has been ignored, although neurons are very noisy devices, as we emphasized in Chapter 2. Synaptic noise introduces undeterminism in the dynamics of a network whose parameters (its sets of interactions and delays) are well-defined quantities. One is then compelled to carry out averages on ensembles of networks which undergo different uncorrelated sources of synaptic noises. This thermal randomness is present in all types of dynamics. In Glauber dynamics thermal randomness combines with delay randomness, whereas in Little dynamics thermal randomness is the sole source of noise.

- a) *Algorithms for neural dynamics with thermal noise.* — We first indicate how the dynamics of neural networks transform when noise is taken into account. Because the equations prove to be easier to handle with symmetrical coding $\sigma_i \in \{-1, +1\}$ than with binary coding $S_i \in \{0, 1\}$, the symmetrical coding will be used henceforth.

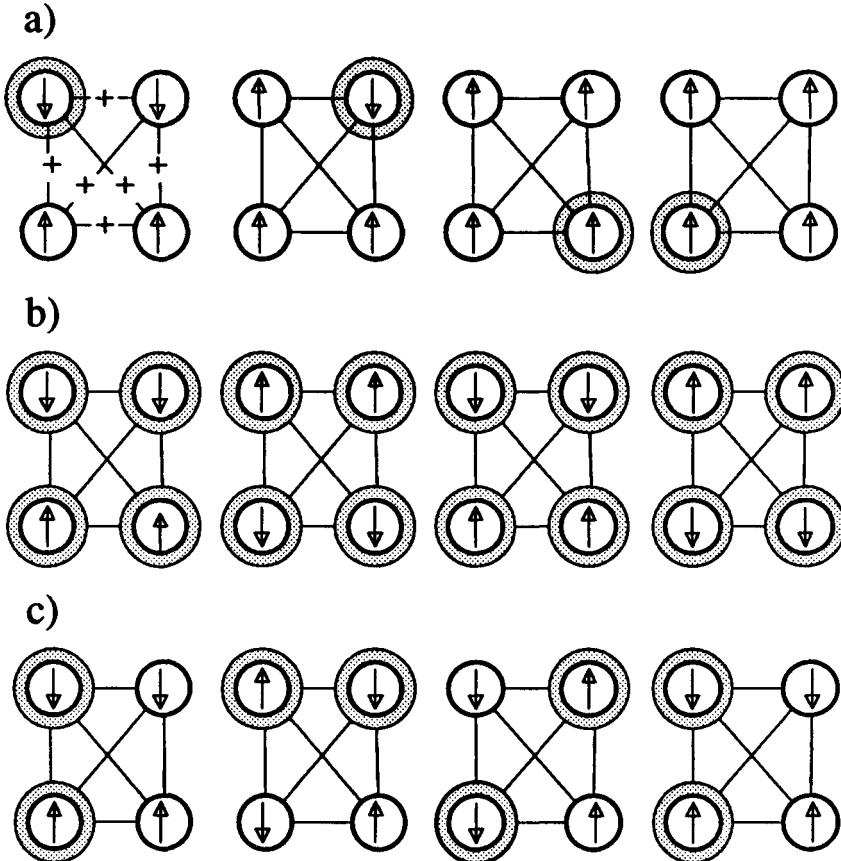


Figure 3.4. Attractors in a 4-unit symmetrical network. Shaded circles show updated neurons. $J_{ij} = +1 (\forall i, j)$.

a) Serial dynamics ($N^b = 1$). The attractor is a fixed point.

b) Parallel dynamics ($N^b = 4$). The attractor is a limit cycle of length 2.

c) Block dynamics ($N^b = 2$). The attractor is a limit cycle of length 3.

We have seen in section 3.1 that the probability for a neuron i to be in state $\sigma_i = 1$ is given by

$$P[\sigma_i = 1] = 1 - P[\sigma_i = -1] = \mathcal{S}^0(\beta h_i). \quad (3.34)$$

These two equations can be combined, thus leading to

$$P[\sigma_i] = \frac{1}{2} [1 + \sigma_i(2\mathcal{S}^0(\beta h_i) - 1)] = \frac{1}{2} [1 + \sigma_i \mathcal{S}(\beta h_i)], \quad (3.35)$$

where the identity $\sigma_i \equiv 2S_i - 1$ has been used.

The Glauber algorithm of noisy networks is rewritten accordingly

- 1) Choose a neuron i at random.
- 2) Compute the mean field

$$h_i(\nu - 1) = \sum_j J_{ij} \sigma_j(\nu - 1) - \theta_i.$$
- 3) Update the state $\sigma_i(\nu) = \mp 1$ of neuron i
with probability

$$P(\sigma_i(\nu)) = \frac{1}{2}[1 + \sigma_i \mathcal{S}(\beta h_i(\nu - 1))].$$
- 4) Iterate in 1).

The Glauber algorithm for noisy networks.

It is equivalent, but of much more practical interest, to replace step 3 by one which is simply adding a random contribution η to the field h_i and comparing the result with zero. The distribution $P(\eta)$ of η must be given by

$$P(\eta) = \left. \frac{d\mathcal{S}(\beta x)}{dx} \right|_{x=\eta} \quad (3.36)$$

for the two processes to display the same statistical properties. For example the noise distribution is

$$P(\eta) = \frac{\beta}{\cosh^2(\beta \eta)}$$

for thermodynamical noises and

$$P(\eta) = \frac{1}{\sqrt{2\pi\beta^2}} \exp\left(-\frac{\eta^2}{2\beta^2}\right)$$

for Gaussian noises.

The Glauber algorithm becomes

- 1) Choose a neuron i at random.
- 2) Compute its mean field

$$h_i(\nu - 1) = \sum_j J_{ij} \sigma_j(\nu - 1) - \theta_i.$$
- 3) Add a noise η , whose distribution probability
is given by Eq. (3.36), to the field h_i .
- 4) Update the state of neuron i according to

$$\sigma_i(\nu) = \text{sign}(h_i(\nu - 1) + \eta).$$
- 5) Iterate in 1)

A new version of the Glauber algorithm for noisy networks.

The Little dynamics may be transformed in a similar way

- 1) Compute the fields h_i for all neurons i .
 - 2) Add random noises η_i , obeying the distribution (3.36) to every field h_i .
 - 3) Update simultaneously all states according to:
- $$\sigma_i(\nu) = \text{sign}(h_i(\nu - 1) + \eta_i).$$
- 4) Iterate in 1).

A Little algorithm for noisy networks.

b) *Markov chains.* — The history of a given system depends on the string of noises η it has experienced in the past. Following the methodology of statistical mechanics, one considers an *ensemble of systems* Σ^s whose dynamics obey, one or the other of the above algorithms. The systems Σ^s of the ensemble have all the same sets $\{J_{ij}\}$ of interactions but their respective ‘noise histories’, that is, the respective strings of random fields η_i , are different. These strings, however, have to obey the constraint C that their distributions, characterized by a zero mean value and a variance $B = \beta^{-1}$, are the same. The averages of some observables O carried out on ensembles of systems with different noise histories are called *thermal averages*. To distinguish them from *averages over realizations* \bar{O} , which have been introduced in the last section, thermal averages are denoted $\langle O(\nu) \rangle$ (see Fig. 3.5). They are given by Eq. (3.24):

$$\langle O(\nu) \rangle = \sum_I \rho(I, \nu) O(I),$$

where, as usual, $\rho(I, \nu)$ is the *probability* that a network of the ensemble is in state I at time ν .

The vector $\tilde{\rho}(\nu) = \{\rho(I, \nu)\}$ is normalized,

$$\sum_I \rho(I, \nu) = 1,$$

and the time evolution of the distribution probability is determined by a transition matrix \mathbf{W} whose elements $W(I | J)$ are the probability for the network to be in state I at time ν , knowing that it is in state J at time $(\nu - 1)$. \mathbf{W} is a statistical matrix,

$$\sum_I W(I | J) = 1,$$

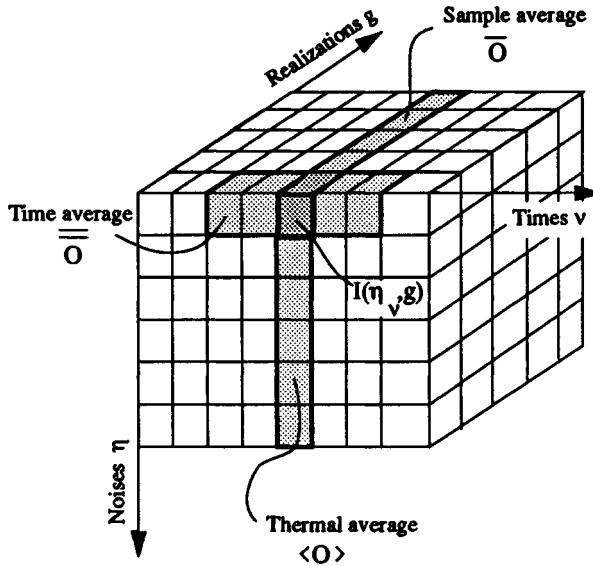


Figure 3.5. Three sorts of ensemble averagings. Time and thermal averagings and averaging over realizations.

and the distribution probability obeys a master equation,

$$\tilde{\rho}(\nu) = \mathbf{W} \cdot \tilde{\rho}(\nu - 1).$$

Given $\tilde{\rho}(\nu = 0)$, the initial distribution of states, the master equation yields

$$\tilde{\rho}(\nu) = \mathbf{W}^\nu \cdot \tilde{\rho}(\nu = 0) \quad (3.37)$$

for the distribution at time ν . The set of successive vectors $\tilde{\rho}(\nu)$ makes up a Markov chain.

c) *The transition matrices.* — The elements of the transition matrices \mathbf{W} associated with the Glauber and those associated with the Little dynamics can be made explicit. We define $w_i(\sigma_i | J)$ as the probability for neuron i to be in state σ_i at time ν , given the lattice is in state J at time $(\nu - 1)$. This probability is

$$w_i(\sigma_i | J) = \frac{1}{2} [1 + \sigma_i \mathcal{S}(\beta h_i(J))], \quad (3.38')$$

with

$$h_i(J) = \sum_j J_{ij} \sigma_j(J) - \theta_i. \quad (3.38'')$$

The target state I is the set of all states $\sigma_i : I = \{\sigma_i\}$ and $W(I | J)$ is the probability that $\sigma_i = \sigma_i(I)$, ($\forall i$), given J . As there are no restrictions on the target state in the Little dynamics, the elements of its associated transition matrix are

$$W(I | J) = \prod_{i=1}^N w_i(\sigma_i | J) = \prod_i^N \frac{1}{2} [1 + \sigma_i(I) \mathcal{S}(\beta h_i(J))]. \quad (3.39)$$

In the Glauber dynamics, on the other hand, the target states which one can access to starting from state J are the states I which are in the neighborhood $\mathcal{V}(J)$ of J . It must be reminded that the neighborhood $\mathcal{V}(J)$ of J is defined by

$$\begin{aligned} I \in \mathcal{V}(J) &\quad \text{if } (I \equiv J) \\ &\quad \text{or } [\sigma_k(I) = \sigma_k(J) \text{ for } k \neq i \text{ and } \sigma_i(I) = -\sigma_i(J)]. \end{aligned}$$

Therefore the off-diagonal elements of the transition matrix associated to the dynamics of Glauber are

$$W(I | J) = \begin{cases} \frac{1}{N-1} w_i(\sigma_i | J) & \\ = \frac{1}{2(N-1)} [1 + \sigma_i(I) \mathcal{S}(\beta h_i(J))] & \text{if } I \in \mathcal{V}(J), \\ 0 & \text{if } I \notin \mathcal{V}(J), \end{cases} \quad (3.40')$$

and the diagonal elements are given by

$$W(J | J) = 1 - \sum_{I \neq J} W(I | J). \quad (3.40'').$$

Equations (3.37) and (3.39) or (3.40) determine the dynamics of neural networks.

3.3.2 The steady distributions of states

In this section we mention some important properties of Markovian systems. For example:

a) The right Perron eigenvector

This is the steady distribution of states. Let

$$\lambda_n, \tilde{\rho}_n^T, \tilde{\rho}_n^\ell, \quad n = 1, 2, \dots, 2^N,$$

be the eigenvalues and the associated right and left eigenvectors of matrix \mathbf{W} . One writes

$$\mathbf{W}^\nu = \sum_{n=1}^{2^N} \lambda_n^\nu \tilde{\rho}_n^r \cdot (\tilde{\rho}_n^l)^T$$

and
$$\tilde{\rho}(\nu) = \sum_{n=1}^{2^N} \lambda_n^\nu ((\tilde{\rho}_n^l)^T \cdot \tilde{\rho}(\nu = 0)) \cdot \tilde{\rho}_n^r.$$

Let us study the asymptotic distribution of states. As ν goes to infinity all contributions of the various terms n vanish except that arising from the Perron eigenvalue $\lambda_1 = 1$. This is related to the fact that in Glauber dynamics (and in noisy enough Little dynamics) one is certain that the Perron eigenvector is the sole eigenvalue with a module of 1 (the proof is given below, along with some results regarding the eigenvalues of \mathbf{W}). On the other hand, since

$$(\tilde{\rho}_1^l)^T = (1, 1, 1, 1, \dots),$$

one has $(\tilde{\rho}_1^l)^T \cdot \tilde{\rho}(\nu = 0) = \sum_I \rho(I, \nu = 0) = 1$ and therefore

$$\tilde{\rho}(\nu \rightarrow \infty) = \tilde{\rho}^* = \tilde{\rho}_1^r.$$

The conclusion is that the right Perron eigenvector $\tilde{\rho}_1^r$ is the asymptotic (stationary) distribution of states.

A few remarks on the distribution of eigenvalues of matrix \mathbf{W}

One first remarks that $\lambda_1 = 1$ is always an eigenvalue of statistical matrices such as \mathbf{W} . This special eigenvalue is called the Perron eigenvalue. The corresponding eigenvectors are $(\tilde{\rho}_1^l)^T = (1, 1, \dots, 1)$, the left eigenvector, and $\tilde{\rho}_1^r$, the right eigenvector.

On the other hand it is worth noting that all the (generally complex) eigenvalues of a statistical matrix \mathbf{W} are in a disk \mathcal{D} of radius 1 centered at $0 + i \times 0$, the origin of the complex plane. This can be proved by using the Gershgorin theorem:

All the eigenvalues λ_n of a (general) matrix \mathbf{W} are in the union of the disks defined by

$$|W(I | I) - \lambda_n| = \sum_{J \neq I} |W(I | J)|,$$

and in the union of the disks defined by

$$|W(I | I) - \lambda_n| = \sum_{J \neq I} |W(J | I)|.$$

Proof. — Let λ_n be an eigenvalue of \mathbf{W} , $\tilde{\rho}_n^r$ be the corresponding right eigenvector and $\rho_n(I)$ the component of largest module of $\tilde{\rho}_n^r$. Then

$$\begin{aligned} |W(I \mid I) - \lambda_n| \times |\rho_n(I)| &= |(W(I \mid I) - \lambda_n)\rho_n(I)| \\ &= \left| W(I \mid I)\rho_n(I) - \sum_J W(I \mid J)\rho_n(J) \right| \\ &= \left| \sum_{J \neq I} W(I \mid J)\rho_n(J) \right| \\ &< \sum_{J \neq I} |W(I \mid J)| \times |\rho_n(I)|. \end{aligned}$$

This proves the first part of the theorem; the proof of the other part uses the transposed matrix \mathbf{W}^T instead of \mathbf{W} .

To show that all eigenvalues of \mathbf{W} lie in \mathcal{D} we note that the center of one of the disks that makes up \mathcal{D} is on the real axis at distance $W(I \mid I)$ from origin. Its radius, owing to the normalization of columns of \mathbf{W} , is less than or equal to $1 - W(I \mid I)$. Therefore the disk is necessarily included in \mathcal{D} . We note also that, according to the results of section 3.2.2, the eigenvalues of noiseless networks either are zero or are roots of 1.

There is *a priori* no restriction on the location in the complex plane of the eigenvalues of the transition matrix \mathbf{W} of Little dynamics except that they must lie in \mathcal{D} . We remember that in noiseless networks \mathbf{W} can comprise a number of cyclic submatrices. Each cyclic submatrix is associated with a limit cycle of length ℓ which generates ℓ th-roots of 1. As soon as noise comes into play non-zero transition elements appear in \mathbf{W} between states of a same cycle or between states of different cycles. This destroys the normalization of most eigenvalues which, therefore, lie strictly inside \mathcal{D} (except the generally much degenerated Perron eigenvalue).

There is no guarantee that *all* eigenvalues (ignoring the Perron eigenvalue) of the Little matrix \mathbf{W} strictly lie inside \mathcal{D} . For the Glauber dynamics, however, the modules of all eigenvalues of \mathbf{W} , except the Perron eigenvalue, are strictly smaller than 1, provided that the self-interactions J_{ii} vanish. As a matter of fact, we shall prove that the eigenvalues of the transition matrix associated to a Glauber dynamics are in a disk \mathcal{D}' of radius 0.5 centered at $0.5 + i \times 0$.

Proof. — If $J_{ii} = 0$, one notes that

$$h_i(I) = h_i(J),$$

since in the Glauber dynamics the states I and J are identical except on site i :

$$\sigma_i(I) = -\sigma_i(J).$$

Therefore, according to Eq. (3.40), one has

$$W(I \mid J) + W(J \mid I) = \frac{1}{N-1}; \quad (I \neq J)$$

and

$$\sum_{I \neq J} (W(I \mid J) + W(J \mid I)) = 1.$$

Using the Gershgorin theorem to the J th line and to the J th column of \mathbf{W} , one finds that the radius of the disk associated with the diagonal element J is given by

$$\lambda_J = \min \left\{ \sum_{I \neq J} W(I | J) ; \sum_{I \neq J} W(J | I) \right\}.$$

If $W(J | J) > \frac{1}{2}$, then λ_J is given by $1 - W(J | J) < \frac{1}{2}$. If $W(J | J) < \frac{1}{2}$, then λ_J is given by $1 - (1 - W(J | J)) = W(J | J) < \frac{1}{2}$, which establishes the statement.

b) *The detailed balance principle*

We prove that if the elements $W(I | J)$ of the matrix transition obey the relation

$$\frac{W(I | J)}{W(J | I)} = \frac{F(I)}{F(J)}, \quad (3.41)$$

then the elements of stationary distribution $\tilde{\rho}^*$ are given by

$$\rho^*(I) = \frac{F(I)}{Z}, \quad Z = \sum_J F(J). \quad (3.42)$$

The conditions (3.41) are called the *detailed balance conditions* and Z is called the *partition function*.

Proof. — One computes

$$\begin{aligned} \sum_J W(I | J)F(J) &= \sum_J \frac{W(I | J)W(J | I)}{W(I | J)} F(I) \\ &= F(I) \sum_J W(J | I) = F(I). \end{aligned}$$

Therefore the vector with components $F(I)$ is an eigenvector of \mathbf{W} with $\lambda = 1$. This is the asymptotic distribution $\tilde{\rho}^*$. The denominator Z in $\rho^*(I)$ is added for the sake of normalization.

c) *The limit distribution in the Glauber dynamics*

In this subsection, the sigmoid function $S(x)$ is explicitly given the form of a tanh function: $S(x) = \tanh(x)$. The elements of \mathbf{W} in the asynchronous dynamics are

$$W(I | J) = \frac{1}{2N} \left(1 + \tanh(\beta \sigma_i(I) h_i(J)) \right),$$

where the symmetry of the tanh function is used. To go further it is necessary to assume that the interactions are *symmetrical* and that *self-connections are absent*. According to the results of section 3.2.5 an energy function $H(I)$ given by

$$H(I) = - \sum_{\langle ij \rangle} J_{ij} \sigma_i(I) \sigma_j(I) + \sum_i \theta_i \sigma_i(I)$$

is introduced and the argument of the tanh function can be written as

$$2\sigma_i(I)h_i(J) = H(J) - H(I).$$

Using the identity $\frac{1}{2}(1 + \tanh(x)) \equiv \frac{e^x}{2\cosh(x)}$, this expression can be transformed into

$$W(I | J) = \frac{\exp \frac{1}{2}\beta[(H(J) - H(I))]}{2N \cosh[\frac{1}{2}\beta(H(J) - H(I))]}$$

or

$$\frac{W(I | J)}{W(J | I)} = \frac{\exp -\beta H(I)}{\exp -\beta H(J)},$$

which obeys the detailed balance principle and therefore yields the asymptotic distribution $\rho^*(I)$:

$$\rho^*(I) = \frac{1}{Z} \exp\left[-\frac{H(I)}{B}\right], \quad Z = \sum_J \exp\left[-\frac{H(J)}{B}\right]. \quad (3.43)$$

This is the Maxwell Boltzmann distribution of states of statistical mechanics: symmetrically connected neural networks are then strictly equivalent to a class of systems, the Ising spin systems, which has been the focus of intensive research in solid-state physics. Spin glasses are magnetic systems with random symmetrical interactions. It is tempting to apply to symmetrically connected networks the tools and the concepts which have been devised for spin glasses.

d) The limit distribution in the Little dynamics

The elements of the transition matrix of the Little dynamics are given by

$$W(I | J) = \prod_{i=1}^N \frac{\exp +\beta \sigma_i(I) h_i(J)}{2 \cosh(\beta h_i(J))} = \frac{\exp +\beta \sum_{i,j} J_{ij} \sigma_i(I) \sigma_j(J)}{2^N \prod_i \cosh(\beta h_i(J))}.$$

If the interactions are *symmetrical*, one has

$$\exp +\beta \sum_{i,j} J_{ij} \sigma_i(I) \sigma_j(J) = \exp +\beta \sum_{i,j} J_{ij} \sigma_i(J) \sigma_j(I)$$

and therefore

$$\frac{W(I | J)}{W(J | I)} = \frac{\prod_i \cosh(\beta h_i(I))}{\prod_i \cosh(\beta h_i(J))}.$$

The elements obey the detailed balance conditions and the state distribution is given by

$$\rho^*(I) = \frac{1}{Z} \prod_{i=1}^N \cosh(\beta h_i(I)) \quad \text{with} \quad Z = \sum_J \prod_{i=1}^N \cosh(\beta h_i(J)).$$

This can be thought as the stationary distribution of a system, with an energy function given by

$$\begin{aligned} H(I) &= -\frac{1}{\beta} \log(Z\rho^*(I)) \\ &= -\frac{1}{\beta} \sum_{i=1}^N \log \left[\cosh \left(\beta \sum_j J_{ij} \sigma_j(I) \right) \right]. \end{aligned} \quad (3.44)$$

Remark

As a matter of fact, Gaussian noise seems to be more realistic than thermal noise. Starting with error functions instead of tanh response functions, it is not possible to express the steady distribution $\tilde{\rho}^*(I)$ in a simple way. It must be noted however that a tanh function can be made similar to an erf function within a maximum error range of 1% by choosing convenient noise parameters β . One expects that properties of biological systems are robust with respect to changes of that sort. It is therefore assumed that the analysis of the relevant properties of neural networks can be carried out by using the former type of response function.

3.3.3 The dynamics of ‘instantaneous’ average activities

We now derive the equation driving the thermal averaged activity $\langle \sigma_i \rangle$ of neural networks for first-order dynamics, that is for Glauber as well as for Little dynamics.

- a) In section 3.3.1 we have defined the thermal average value at time ν of an observable O by

$$\langle O \rangle_\nu = \sum_I O(I) \rho(I, \nu).$$

In particular the ensemble average of σ_i yields the time dependence of the instantaneous frequency of neuron i ,

$$\langle \sigma_i \rangle_\nu = \sum_I \sigma_i(I) \rho(I, \nu).$$

Using the master equation one can write

$$\begin{aligned} \langle \sigma_i \rangle_{\nu+1} &= \sum_I \sigma_i(I) \rho(I, \nu + 1) \\ &= \sum_I \sigma_i(I) \left(\sum_J W(I | J) \rho(J, \nu) \right) \\ &= \sum_J \left(\sum_I W(I | J) \sigma_i(I) \right) \rho(J, \nu), \end{aligned}$$

which, with the definition of ensemble averages, is written as

$$\langle \sigma_i \rangle_{\nu+1} = \left\langle \sum_I W(I | J) \sigma_i(I) \right\rangle_\nu.$$

We let $F_i(J) = \sum_I W(I | J) \sigma_i(I)$, whence

$$\langle \sigma_i \rangle_{\nu+1} - \langle \sigma_i \rangle_\nu = -(\langle \sigma_i \rangle_\nu - \langle F_i \rangle_\nu), \quad (3.45')$$

or, in differential form,

$$\frac{d\langle \sigma_i \rangle}{dt} = -\frac{1}{\tau} (\langle \sigma_i \rangle - \langle F_i \rangle). \quad (3.45'')$$

b) An explicit expression for $F_i(J)$ can be derived by using Eqs (3.39) and (3.40). When the dynamics is synchronous, the transition matrix element $W(I | J)$ is given by

$$W(I | J) = \prod_{i'=1}^N \frac{1}{2} [1 + \sigma_{i'}(I) \mathcal{S}(\beta h_{i'}(J))] \quad (3.46)$$

and

$$\begin{aligned} F_i(J) &= \sum_I W(I | J) \sigma_i(I) \\ &= \sum_I \frac{1}{2} [\sigma_i(I) + \mathcal{S}(\beta h_i(J))] \prod_{i' \neq i} \frac{1}{2} [1 + \sigma_{i'} \mathcal{S}(\beta h_{i'}(J))]. \end{aligned}$$

Taking the following equalities into account

$$\begin{aligned} \sum_{\sigma_i \in \{+1, -1\}} \frac{1}{2} [\sigma_i(I) + \mathcal{S}(\beta h_i(J))] &= \mathcal{S}(\beta h_i(J)); \\ \sum_{\sigma_{i'} \in \{+1, -1\}} \frac{1}{2} [1 + \sigma_{i'}(I) \mathcal{S}(\beta h_{i'}(J))] &= 1 \end{aligned}$$

(provided that the neurons do not self connect, $J_{i'i'} = 0$), one finds

$$F_i(J) = \mathcal{S}(\beta h_i(J))$$

and

$$\frac{d\langle \sigma_i \rangle}{dt} = -\frac{1}{\tau} [\langle \sigma_i \rangle - \langle \mathcal{S}(\beta h_i) \rangle]. \quad (3.47)$$

Similarly for the asynchronous dynamics

$$\begin{aligned} W(I | J) &= \frac{1}{2(N-1)} [1 + \sigma_i(I) \mathcal{S}(\beta h_i(J))], \quad (I \neq J); \\ W(J | J) &= 1 - \sum_{I \neq J} \frac{1}{2(N-1)} [1 + \sigma_i(I) \mathcal{S}(\beta h_i(J))], \end{aligned} \quad (3.48)$$

where the state J is a neighbor of the state I on site i . Therefore

$$\begin{aligned} F_i(J) &= \sum_I W(I | J) \sigma_i(I) \\ &= W(J | J) \sigma_i(J) + \sum_{I \neq J} W(I | J) \sigma_i(I), \end{aligned}$$

$$\begin{aligned} \text{and } F_i(J) &= \sigma_i(J) - \sum_{I \neq J} \frac{1}{2(N-1)} [1 + \sigma_i(I) \mathcal{S}(\beta h_i(J))] \sigma_i(J) \\ &\quad + \cdots + \sum_{I \neq J} \frac{1}{2(N-1)} [1 + \sigma_i(I) \mathcal{S}(\beta h_i(J))] \sigma_i(I) \\ &= \sigma_i(J) - \left[\sum_{I \neq J} \frac{1}{N-1} \right] [\sigma_i(J) - \mathcal{S}(\beta h_i(J))]. \end{aligned}$$

Since there is only one state J that is a neighbor of state I at site i , one has

$$\sum_{I \neq J} \frac{1}{N-1} = \frac{1}{N-1},$$

and, from Eq. (3.45''), one finally finds

$$\frac{d\langle \sigma_i \rangle}{dt} = -\frac{1}{\tau_r} [\langle \sigma_i \rangle - \langle \mathcal{S}(\beta h_i) \rangle], \quad (3.49)$$

since $(N-1)\tau \simeq \tau_r$.

c) Remarks

- The equations (3.49) are *rigorous*. Their derivation appeals to no assumption other than the absence of self-connections.
- The equations (3.49) are quite general. In particular:
 - They are valid for any response function $\mathcal{S}(x)$, even those with no special symmetry properties.
 - They do not depend on the dynamics, which enters only through the conditional probabilities $w_i(\sigma_i | I)$. Therefore the results they possibly yield are valid for any type of block dynamics, for Glauber or for Little dynamics for example. The type of dynamics only fixes the time scale τ of the evolution. An important consequence is that the average properties of the systems are dynamics-independent in spite of the seemingly different forms that the steady distributions can take as in equations (3.43) or (3.44).
- When the variables S_i are used instead of variables σ_i , the change of variables $S_i = \frac{1}{2}(\sigma_i + 1)$ yields

$$\frac{d\langle S_i \rangle}{dt} = -\frac{1}{\tau} (\langle S_i \rangle - \langle \mathcal{S}^0(\beta h_i) \rangle),$$

where $\mathcal{S}^0(x)$ is related to $\mathcal{S}(x)$ by Eq. (3.7).

3.3.4 A direct derivation of the equations of motion

We give here a more ‘visual’ derivation of the dynamical equations (3.49). The calculations are carried out in the framework of the Glauber (asynchronous) dynamics. There is essentially nothing new here but the technique that is used can be easily transposed to the calculations of more involved observables such as that of correlation functions, for example.

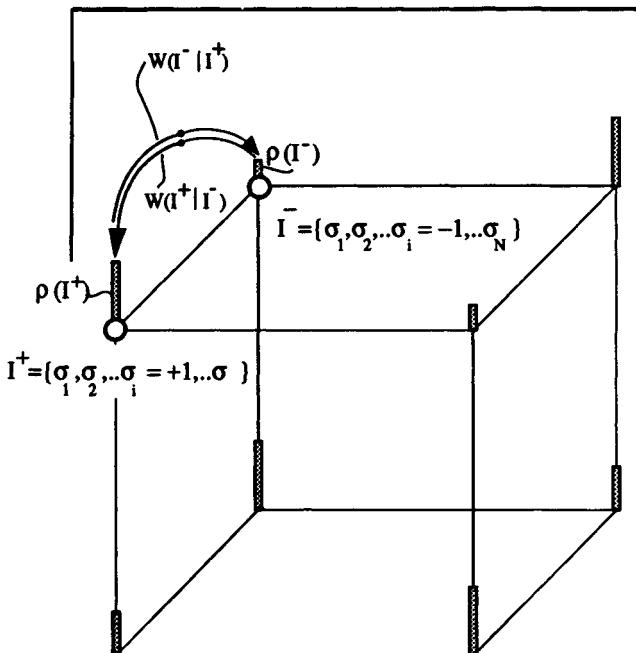


Figure 3.6. The evolution dynamics in the phase space of a three-unit neural network.

The probabilities $\rho(I, \nu)$ can be viewed as stacks of tokens that are attached at time ν to the various summits I of the N -dimensional hypercube representing the phase space of the network (see Fig. 3.6). There are as many tokens on vertex I as systems of the averaging ensemble which are in state I at time ν . Let i be the neuron to be updated at time ν . We define

$$I^+ = (\sigma_i, \sigma_2, \dots, \sigma_i = +1, \dots, \sigma_N)$$

and

$$I^- = (\sigma_1, \sigma_2, \dots, \sigma_i = -1, \dots, \sigma_N).$$

The states I^+ and I^- are on either side of a plane P_i which splits the hypercube into two halves. P_i is traversed by one token each time the state σ_i of a neuron i of a

given system of the ensemble flips. Counting at time ν , the difference in the numbers of tokens on each side of \mathcal{P}_i gives the average activity of neuron i :

$$\langle \sigma_i \rangle_\nu = \sum_{I^+} \rho(I^+, \nu) - \sum_{I^-} \rho(I^-, \nu).$$

The dynamics of the activity is accounted for by the number of tokens crossing \mathcal{P}_i in one time step. This yields

$$\begin{aligned} \langle \sigma_i \rangle_{\nu+1} - \langle \sigma_i \rangle_\nu &= -2 \sum_{I^+} w_i(\sigma_i = -1 | I^+) \rho(I^+, \nu) \\ &\quad + 2 \sum_{I^-} w_i(\sigma_i = +1 | I^-) \rho(I^-, \nu), \end{aligned}$$

where, for example, $w_i(\sigma_i = -1 | I^+)$ is the probability that σ_i flips to $\sigma_i = -1$, given that the state of the network is $I^+ = \{I'\} \otimes \{\sigma_i = +1\}$. This probability is given by the matrix element of the Glauber transition matrix \mathbf{W} in Eqs (3.38):

$$w_i(\sigma_i = -1 | I^+) = \frac{1}{2(N-1)} \left[1 - \mathcal{S}(\beta h_i(I^+)) \right].$$

We also have

$$w_i(\sigma_i = +1 | I^-) = \frac{1}{2(N-1)} \left[1 + \mathcal{S}(\beta h_i(I^-)) \right]$$

with $I^- = \{I'\} \otimes \{\sigma_i = -1\}$.

$$\begin{aligned} \langle \sigma_i \rangle_{\nu+1} - \langle \sigma_i \rangle_\nu &= -\frac{1}{N-1} \sum_{I^+} \left[1 - \mathcal{S}(\beta h_i(I^+)) \right] \rho(I^+, \nu) \\ &\quad + \frac{1}{N-1} \sum_{I^-} \left[1 + \mathcal{S}(\beta h_i(I^-)) \right] \rho(I^-, \nu) \\ &= -\frac{1}{N-1} \sum_I \sigma_i(I) \left[1 - \sigma_i(I) \mathcal{S}(\beta h_i(I)) \right] \rho(I, \nu) \\ &= -\frac{1}{N-1} \left[\langle \sigma_i \rangle_\nu - \langle \mathcal{S}(\beta h_i) \rangle_\nu \right] \end{aligned}$$

since $(\sigma_i)^2 = 1$. In the continuous time limit this equation becomes

$$\frac{d\langle \sigma_i \rangle}{dt} = -\frac{1}{(N-1)\tau} \left[\langle \sigma_i \rangle - \langle \mathcal{S}(\beta h_i) \rangle \right] = -\frac{1}{\tau_r} \left[\langle \sigma_i \rangle - \langle \mathcal{S}(\beta h_i) \rangle \right]. \quad (3.49')$$

One recovers Eq. (3.49).

Remarks

- The derivation does not appeal to the hypothesis of non-self-connectivity (but it is limited to the Glauber dynamics).
- It provides the time scale of the dynamics: the time constant is simply the refractory period τ_r .

- It can easily be extended to that of the dynamics of correlation functions such as $\langle \sigma_i \sigma_j \rangle$. It is then necessary to consider two planes \mathcal{P}_i and \mathcal{P}_j which divide the phase space in four regions and to calculate the flow of tokens between all regions. One finds

$$\frac{d\langle \sigma_i \sigma_j \rangle}{dt} = -\frac{1}{\tau_r} \left[\langle \sigma_i \sigma_j \rangle - \frac{1}{2} \left(\langle \sigma_j S(\beta h_i) \rangle + \langle \sigma_i S(\beta h_j) \rangle \right) \right].$$

This type of equation could be useful if it appears that correlations are important in the coding of neural activities.

3.3.5 Approximate evolution equations for local fields: Hopfield dynamics for analog neural networks

We start from Eq. (3.49) and we make use of a non-trivial assumption which is very common to statistical physics: *the mean field approximation*.[†] The approximation consists in writing

$$\langle S(x) \rangle = S(\langle x \rangle)$$

and therefore

$$\frac{d\langle \sigma_i \rangle}{dt} = -\frac{1}{\tau_r} \left\{ \langle \sigma_i \rangle - S \left(\beta \left[\sum_j J_{ij} \langle \sigma_j \rangle - \theta_i \right] \right) \right\}. \quad (3.50)$$

This provides a set of N coupled equations between the dynamic variables $\langle \sigma_i \rangle$. The equation is inverted:

$$\beta \left(\sum_j J_{ij} \langle \sigma_j \rangle - \theta_i \right) = S^{-1} \left(\langle \sigma_i \rangle + \tau_r \frac{d\langle \sigma_i \rangle}{dt} \right).$$

One assumes, moreover, that the time evolution of $\langle \sigma_i \rangle$ is slow enough to allow a first-order expansion of S^{-1} . Then

$$\beta \left(\sum_j J_{ij} \langle \sigma_j \rangle - \theta_i \right) \simeq S^{-1}(\langle \sigma_i \rangle) + \tau_r \frac{d\langle \sigma_i \rangle}{dt} (S^{-1}(\langle \sigma_i \rangle))'.$$

Finally we introduce a new variable u_i which is proportional to the membrane potential when the system is stationary:

$$u_i = S^{-1}(\langle \sigma_i \rangle).$$

Using

$$\frac{du_i}{dt} = \frac{d\langle \sigma_i \rangle}{dt} (S^{-1}(\langle \sigma_i \rangle))',$$

[†] More details on the mean field approximation are given in section 4.2.4.

the dynamics is given by the following set of coupled equations:

$$\beta^{-1} \tau_r \frac{du_i}{dt} = \left(\sum_j J_{ij} \langle \sigma_j \rangle - \theta_i \right) - \beta^{-1} u_i, \quad (3.51')$$

$$u_i = \mathcal{S}^{-1}(\langle \sigma_i \rangle). \quad (3.51'')$$

These equations are due to Hopfield. Their interest is that they can be implemented in analog electronic circuits: to give a few details a neuron is materialized by an current amplifier with characteristics $i = \mathcal{S}(v)$. The amplifier is connected to an RC circuit (see section 11.1.3). The values of these parameters are given by

$$R = \beta \quad \text{and} \quad C = \beta^{-1} \tau_r,$$

so that $RC = \tau_r$. J_{ij} are resistors connecting amplifier j to amplifier i . With notations adopted by Hopfield

$$V_i = \langle \sigma_i \rangle, \quad T_{ij} = J_{ij}, \quad g(x) = \mathcal{S}(x),$$

his equations take the following form:

$$C \frac{du_i}{dt} = \sum_j T_{ij} V_j - \theta_i - \frac{u_i}{R}, \quad V_i = g(u_i).$$

More information on electronic implementations of neuronal networks is given in Chapter 11.

3.3.6 Experimental confirmations of the stochastic description of neuronal dynamics

It must be admitted that many considerations we have developed rest more on hand-waving arguments than on firmly established theories. It is therefore important to check whether real systems are supported by experimental observations.

Two significant contributions which confirm the main features of the models can be cited here, one by Kleinfeld, Raccuia and Chiel and the other by Thompson and Gibson.

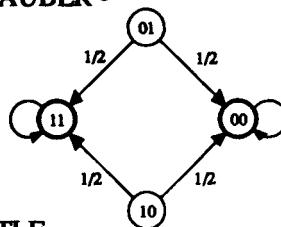
a) Kleinfeld *et al.* build ‘neural circuits in a dish’ as electronics build electronic circuits with transistors. They use identified neurons taken from the abdominal ganglion of a marine mollusc, *Aplysia*, to construct very simple neural networks with definite characteristics, with known synaptic efficacies in particular. More precisely, they make a first circuit consisting of two co-cultured neurons, neurons L7 and L12.

The synaptic contacts between these neurons are excitatory. The authors also build a circuit comprising three neurons with a neuron L10

connected with two so-called LUQ neurons. The two LUQ's make reciprocal inhibitory connections with the L10 cell. Since they are electrically coupled the LUQ neurons can be considered as a single cell from the point of view of the dynamics. The activities are monitored on microelectrodes penetrating the somas of the cells. The initial state of the circuit is controlled by adjusting bias-currents with respect to the thresholds, which amounts to changing temporarily the thresholds themselves.

Reciprocal excitatory interactions - GLAUBER -

$$W_{\text{as}}^+ = \begin{pmatrix} (11) & (01) & (10) & (00) \\ (11) & 1 & 1/2 & 1/2 & 0 \\ (01) & 0 & 0 & 0 & 0 \\ (10) & 0 & 0 & 0 & 0 \\ (00) & 0 & 1/2 & 1/2 & 1 \end{pmatrix}$$



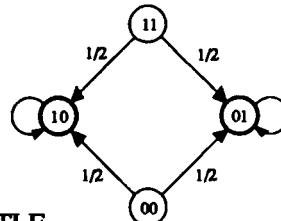
Reciprocal excitatory interactions - LITTLE -

$$W_S^+ = \begin{pmatrix} (11) & (01) & (10) & (00) \\ (11) & 1 & 0 & 0 & 0 \\ (01) & 0 & 0 & 1 & 0 \\ (10) & 0 & 1 & 0 & 0 \\ (00) & 0 & 0 & 0 & 1 \end{pmatrix}$$



Reciprocal inhibitory interactions - GLAUBER -

$$W_{\text{as}}^- = \begin{pmatrix} (11) & (01) & (10) & (00) \\ (11) & 0 & 0 & 0 & 0 \\ (01) & 1/2 & 1 & 0 & 1/2 \\ (10) & 1/2 & 0 & 1 & 1/2 \\ (00) & 0 & 0 & 0 & 0 \end{pmatrix}$$



Reciprocal inhibitory interactions - LITTLE -

$$W_S^- = \begin{pmatrix} (11) & (01) & (10) & (00) \\ (11) & 0 & 0 & 0 & 1 \\ (01) & 0 & 0 & 1 & 0 \\ (10) & 0 & 1 & 0 & 0 \\ (00) & 1 & 0 & 0 & 0 \end{pmatrix}$$



Figure 3.7. Flow charts of two unit neural networks according to the Glauber and the Little dynamics for excitatory and inhibitory reciprocal interactions respectively (for $B = 0$).

A two-neuron network can be in one of the four following states: (1,1), (1,0), (0,1) and (0,0). One must consider the two possible dynamics, that of Glauber which is asynchronous, and the synchronous dynamics of Little for the two networks. The four corresponding (zero-noise) matrices and their associated flow charts are depicted in Fig. 3.7.

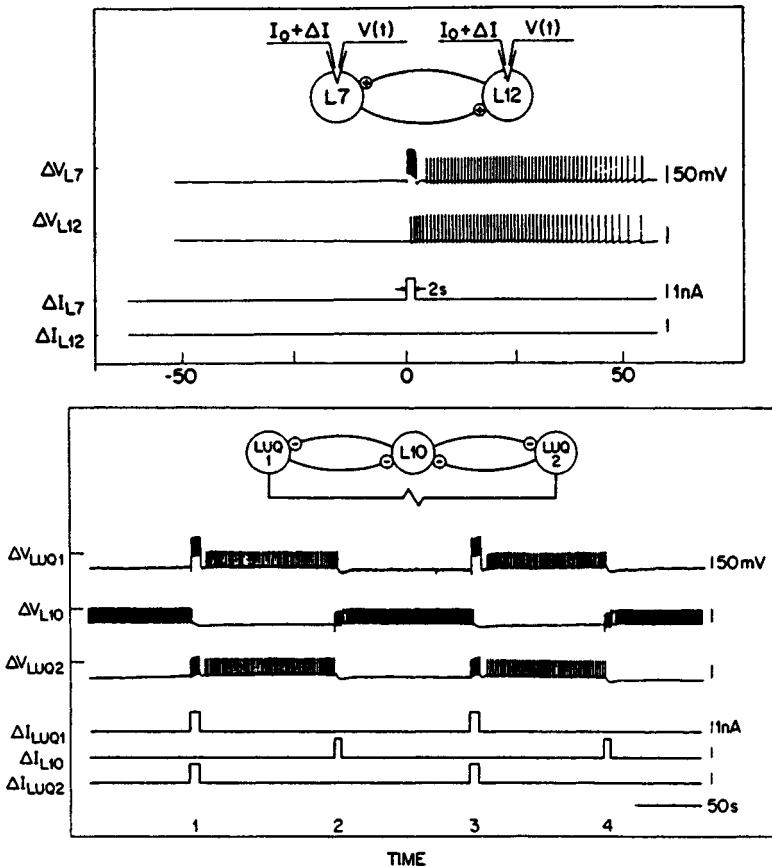
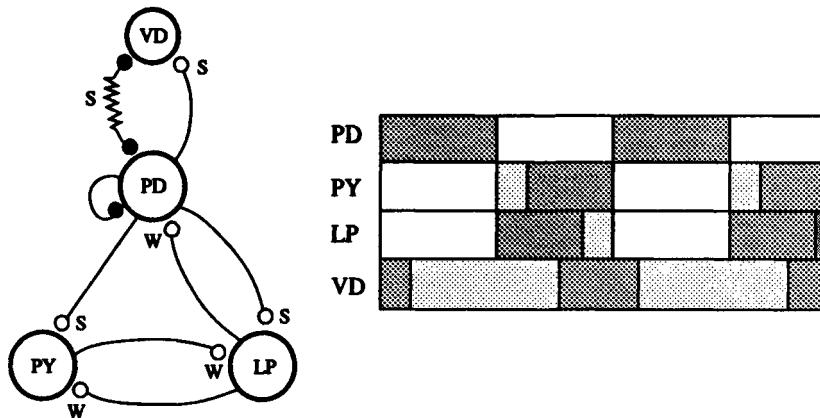


Figure 3.8. Experimental firing patterns observed on two- and three-unit neural networks. The upper figure shows that two neurons related by excitatory synapses are either both active or both silent. The lower figure shows that neurons related by inhibitory synapses have complementary activities (After Kleinfeld *et al.*).

One notes that the Little dynamics is deterministic while the Glauber dynamics remains stochastic even in noiseless networks. Kleinfeld and

a) Biological system



b) The model

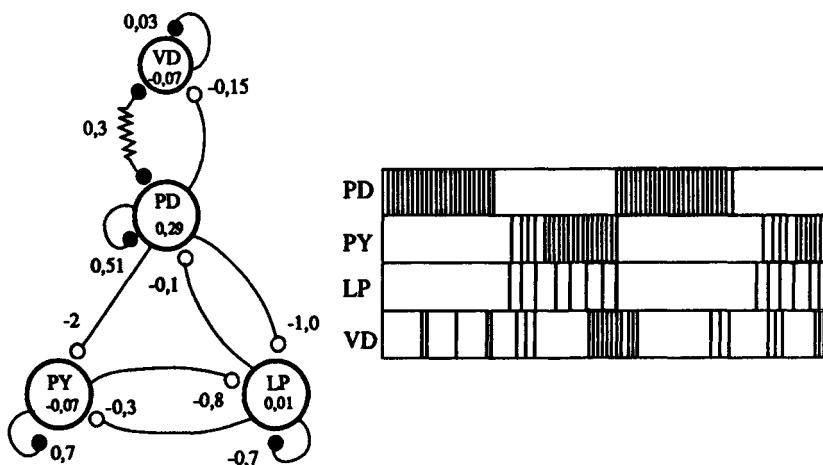


Figure 3.9. Activities of the pyloric system of lobster.

a) The real (simplified) network and its observed activity. S and W are for strong and weak efficacies. Solid and open circles are excitatory and inhibitory synapses respectively. The jagged line is a gap junction.

b) A set of synaptic efficacies and thresholds which fit both the anatomic and the electrophysiological data as shown on the right part of the figure (After Gibson *et al.*).

his co-workers observe that $(1,1)$ and $(0,0)$ are the two fixed points of the dynamics of the first network and $(1,0)$ and $(0,1)$ those of the second network. If the systems are set in one of the fixed points they stay in that state for ever (forgetting the habituation phenomenon). If they are prepared in a transient state by injecting bias current in one of the neurons, the system flips towards one of its fixed points (see Fig. 3.8 taken from Kleinfeld *et al.*). The cyclic behavior $(1,0) \leftrightarrow (0,1) \leftrightarrow (1,0) \dots$, both neurons firing at half maximum frequency, is not observed. All these results are consistent with Glauber dynamics.

b) The experiments of Thompson and Gibson are carried out on a real neural network, the pyloric system of the lobster stomatogastric ganglion. The system is made up of 14 neurons but they are coupled in such a way that the dynamics is identical to that of a 4-neuron system. These neurons are called the pyloric dilatator (PD), the lateral pyloric (LP), the pyloric (PY) and the ventricular dilatator (VD) neurons. All interactions are known at least qualitatively. The network and the observed pattern of activities are displayed in Fig. 3.9.a. In these systems the firing time is a few milliseconds, whereas the delay between a presynaptic spike and the postsynaptic potential is 10 ms. A Little dynamics is therefore more likely than a purely asynchronous one and it is the dynamics which is used for the simulations. The authors show that there exists ranges of values for the parameters of the system, the synaptic efficacies and the thresholds, which are both compatible with the experimental observations and account for the recorded patterns of activities (see Fig. 3.9.b).

The experiments of Kleinfeld and those of Thompson are important because they show that the neuronal dynamics obey simple principles which can be clearly stated and which are embedded in the algorithms we have seen above. It is assumed that the very same principles hold for larger neural networks.

HEBBIAN MODELS OF ASSOCIATIVE MEMORY

Chapter 3 has been devoted to the modeling of the dynamics of neurons. The standard model we arrived at contains the main features which have been revealed by neuroelectrophysiology: the model considers neural nets as networks of probabilistic threshold binary automata. Real neural networks, however, are not mere automata networks. They display specific functions and the problem is to decide whether the standard model is able to show the same capabilities.

Memory is considered as one of the most prominent properties of real neural nets. Current experience shows that imprecise, truncated information is often sufficient to trigger the retrieval of full patterns. We correct misspelled names, we associate images or flavors with sounds and so on. It turns out that the formal nets display these memory properties if the synaptic efficacies are determined by the laws of classical conditioning which have been described in section 2.4. The synthesis in a single framework of observations of neurophysiology with observations of experimental psychology, to account for an emergent property of neuronal systems, is an achievement of the theory of neural networks.

The central idea behind the notion of conditioning is that of associativity. It has given rise to many theoretical developments, in particular to the building of simple models of associative memory which are called Hebbian models. The analysis of Hebbian models has been pushed rather far and a number of analytical results relating to Hebbian networks are gathered in this chapter. More refined models are treated in following chapters.

4.1 Noiseless Hebbian models

4.1.1 *Fixed points and memory*

A given pattern, which will be called the ‘concept’, can be retrieved starting from a wide variety of initial situations, namely ‘percepts’. The concepts are available even in the absence of any external inputs. Therefore they must correspond to some stable states of the neural network. This can be modeled by a system which has the memorized

patterns as fixed points of its dynamics. The percepts are all states belonging to the corresponding basins of attraction (see Fig. 4.1).

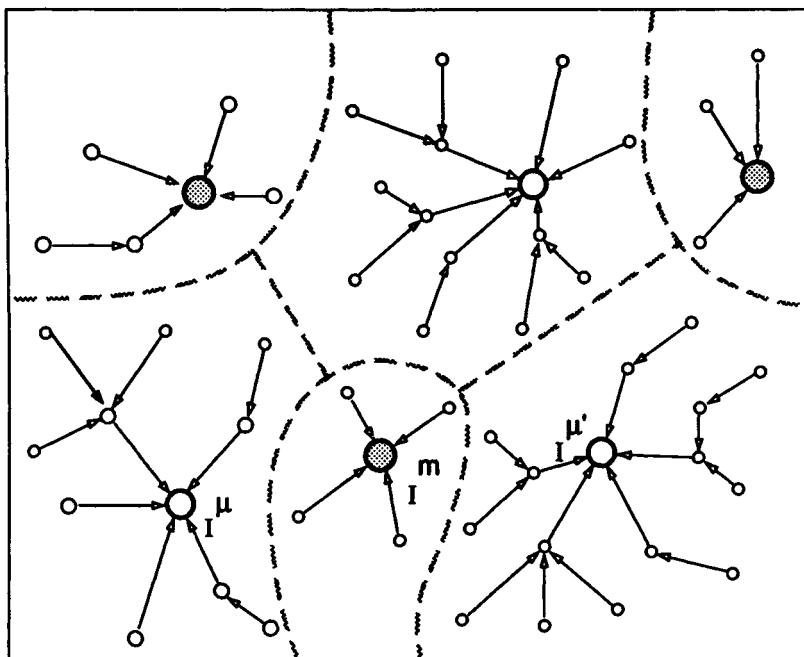


Figure 4.1. The partition of a phase space of a neural network into basins of attraction. Some fixed points such as I^μ are memorized patterns. Others such as I^m are spurious states.

To build a theory of memory it is therefore necessary to:

- devise a neural state dynamics which makes certain that the limit behaviors are fixed points;
- set the connections so as to make the patterns to be memorized the fixed points of the dynamics.

There exist several strategies which meet the first condition. The one considered in this section relies upon the building of an energy function which, as we have seen in the last section, makes sure that the attractors of the dynamics are fixed points.

The second requirement is met by a series of learning algorithms which we shall introduce in the next chapter. In this section only simple Hebbian rules will be considered.

4.1.2 The Hebbian learning rule as an energy minimization process

We have seen in Chapter 2 how the paradigm of association arose from classical conditioning and how the Hebbian form of synaptic dynamics

given in Eq. (2.20) was suggested by these experiments. Here, instead of relying on biological evidence, we look for some form of synaptic efficacies which makes sure that the memorized patterns are the fixed point of the dynamics. We show that these two ways of tackling the problem of memory, the biologically minded one and the automaton minded one, yield the same type of (Hebbian) learning rules.

Let $\{I^\mu\}$ be a set of patterns to be memorized:

$$I^\mu = \{\xi_i^\mu\}, \quad \mu = 1, \dots, P, \quad i = 1, \dots, N, \quad \xi_i^\mu \in \{-1, +1\}.$$

We define P overlaps M^μ as the *projections* of the current state $I = \{\sigma_i\}$ over the patterns I^μ . The set of overlaps forms a P -dimensional vector \widetilde{M} :

$$\widetilde{M} = \{M^\mu\}, \quad M^\mu = \frac{1}{N} \sum_i \xi_i^\mu \sigma_i.$$

We have stressed that memorization requires the patterns I^μ be fixed points of the dynamics. This is achieved by devising an energy function $H(I)$ which is minimum when $I = I^\mu$. Such a function can be built by making the largest overlaps, points of lowest energies in the P -dimensional space spanned by \widetilde{M} :

$$H(I) = -\frac{1}{2} N \sum_{\mu=1}^P (M^\mu)^2. \quad (4.1)$$

Using the definitions of scalar products, the energy can be written as

$$\begin{aligned} H(I) &= -\frac{1}{2N} \sum_\mu \left(\sum_i \xi_i^\mu \sigma_i \right)^2 = -\frac{1}{2N} \sum_i \sum_j \left(\sum_\mu \xi_i^\mu \xi_j^\mu \right) \sigma_i \sigma_j \\ &= -\frac{1}{N} \sum_{\langle ij \rangle} \left(\sum_\mu \xi_i^\mu \xi_j^\mu \right) \sigma_i \sigma_j - \frac{1}{2} P. \end{aligned}$$

This expression is similar to that displayed in Eq. (3.36) except for the constant term $-\frac{1}{2}P$, which plays no role in the dynamics. Identification provides explicit expressions of the parameters of the network:

$$J_{ij} = J_{ji} = \frac{1}{N} \sum_{\mu=1}^P \xi_i^\mu \xi_j^\mu, \quad J_{ii} = 0, \quad \theta_i = 0. \quad (4.2)$$

The efficacies given in Eq. (4.2) can be imagined as the result of a learning dynamics which would modify the synaptic weights by an amount of

$$\Delta J_{ij} = \frac{1}{N} \xi_i^\mu \xi_j^\mu \quad (4.3)$$

when the network experiences a pattern I^μ . This rule is close to that proposed by Hebb: the difference is the substitution of binary variables $S_i^\mu = \frac{1}{2}(\xi_i^\mu + 1)$ by symmetrical variables ξ_i^μ . The symmetrical version of the Hebbian rule has been suggested by Hopfield. It is obvious that the perfect symmetry between the excitatory and the inhibitory interactions, which is inherent to the rule, is not biologically realistic. But the study of this model provides a number of results which, perhaps surprisingly, remain valid when the constraint of symmetrical interactions is removed. This is why it is worth much study. The properties of networks trained with ‘old Hebbian rules’ are examined in section 7.5.5.

4.1.3 Checking the stability of memorized patterns

A pattern I^μ is a fixed point if all local fields $h_i(I^\mu)$ are aligned along the direction of ξ_i^μ for every unit i . Then the coherent influence of all neurons tends strongly to realign a single misoriented unit. Let $I = \{\sigma_i\}$ be a state of the network. The local field on i is written as

$$\begin{aligned} h_i(I) &= \sum_{j=1}^N J_{ij} \sigma_j(I) = \sum_{j=1}^N \left(\frac{1}{N} \sum_{\mu=1}^P \xi_i^\mu \xi_j^\mu \right) \sigma_j(I) \\ &= \sum_{\mu} \frac{1}{N} \left(\sum_j \xi_j^\mu \sigma_j(I) \right) \xi_i^\mu = \sum_{\mu} M^\mu(I) \xi_i^\mu, \end{aligned} \quad (4.4)$$

which shows that every pattern I^μ contributes to the local field by an amount which is proportional to its overlap with the running state I . We now assume that the network is in state $I = I^{\mu_0}$, $\sigma_i = \xi_i^{\mu_0}$, $\forall i$. The sum of fields in Eq. (4.4) is decomposed into a coherent contribution arising from the learning of I^{μ_0} and an incoherent contribution arising from the learning of all the other patterns I^μ , $\mu \neq \mu_0$:

$$\begin{aligned} h_i(I^{\mu_0}) &= \frac{1}{N} \sum_j (\xi_i^{\mu_0} \xi_j^{\mu_0}) \xi_j^{\mu_0} + \frac{1}{N} \sum_{\mu \neq \mu_0} \sum_j (\xi_i^\mu \xi_j^\mu) \xi_j^{\mu_0} \\ &= \xi_i^{\mu_0} + \frac{1}{N} \sum_{\mu \neq \mu_0}^P \sum_j^N \xi_i^\mu \xi_j^\mu \xi_j^{\mu_0}. \end{aligned} \quad (4.5)$$

The state $\sigma_i \equiv \xi_i^{\mu_0}$ is stable if the *stability parameter* $x_i(I^{\mu_0})$ is positive:

$$x_i(I^{\mu_0}) = \xi_i^{\mu_0} h_i(I^{\mu_0}) > 0 \quad (4.6)$$

or

$$x_i(I^{\mu_0}) = 1 + \frac{1}{N} \sum_{\mu \neq \mu_0}^P \sum_j^N \xi_i^\mu \xi_i^{\mu_0} \xi_j^\mu \xi_j^{\mu_0} > 0. \quad (4.7)$$

The first term $x_i^s = 1$ in the r.h.s. of Eq. (3.62') is called the *signal* and the second term x_i^n involving the double sum is called the *noise*. The state $\xi_i^{\mu_0}$ can be destabilized if the noise term becomes larger than the signal term.

One now assumes that the *memorized patterns are random patterns*. The probability distribution of the components of I^μ is then given by

$$P(\xi_i^\mu) = \frac{1}{2} \left(\delta(\xi_i^\mu - 1) + \delta(\xi_i^\mu + 1) \right). \quad (4.8)$$

Noise is then a sum of $N \times P$ random terms, each giving a contribution $+1$ or -1 to the sum. This is the well-known problem of random walk and the result is that noise is a random Gaussian variable with a dispersion of $\sqrt{N \times P}$. A pattern I^μ is stable if

$$1 \geq \frac{1}{N} \sqrt{N \times P} \quad \text{or} \quad P \leq N. \quad (4.9)$$

Condition (4.9) puts an approximative upper limit to the storage capacity of the Hebbian networks.

Remark

- This very crude estimate shows that the memory storage capacity P_C of Hebbian neural networks scales at best as N , the number of neurons. As compared with the total number of states that the network is able to achieve, which is 2^N , this performance looks miserable. The relevant quantity, however, is not the number of stored patterns but the amount of information that is stored. This amount, \mathcal{J}^P , is given by the number of patterns times the number of bits per pattern, $P \times N$. It is to be compared with the number \mathcal{J}^N of bits which is necessary to determine the network itself. A network is efficient if both numbers are the same. In Hebbian neural networks \mathcal{J}^P is of the order of N^2 according to Eq. (4.9). On the other hand Hebbian synapses must be coded on $\log_2 P$ bits since their possible values are $(-P)/N, (-P+2)/N, \dots, (P-2)/N, (P)/N$. The number of synapses is N^2 and therefore \mathcal{J}^N scales as $N^2 \log_2 P$. The shrinking of information storage by a factor of $\log_2 P$ is to be related to the very nature of neuronal dynamics (the majority rule), not to a defect of Hebbian learning.

- Two patterns are orthogonal if their overlap vanishes:

$$M^{\mu\mu'} = \frac{1}{N} \sum_i \xi_i^\mu \xi_i^{\mu'} = 0.$$

The noise term of Eq. (4.5) then disappears and good stability is achieved for all memorized patterns. However, it is not possible for more

than $P = N$ patterns to be each other orthogonal and the maximum memory capacity is still limited to N .

4.1.4 Retrieving patterns without errors

Let us try to destabilize a given pattern, pattern I^1 for example. We start from I^1 and we flip one of its units. The new state, $I^{1,1}$, belongs to the basin of attraction of I^1 if the dynamics of the system brings $I^{1,1}$ back to I^1 . If the case arises we flip one of the units of $I^{1,1}$ and we check whether relaxation brings this new state $I^{1,2}$ back to I^1 and so on till the perturbed state $I^{1,R}$ eventually fails to return to I^1 . This procedure gives an estimate of the size R of the basin associated with I^1 .

When R states have been flipped, the signal term $x_i^s(I^{1,R})$ is given by

$$x_i^s = \xi_i^1 \frac{1}{N} \sum_{j=1}^N (\xi_i^1 \xi_j^1) \sigma_j(I^{1,R}) = x^s = \left(1 - \frac{2R}{N}\right). \quad (4.10)$$

The noise term x_i^n is a Gaussian variable with the following distribution:

$$P(x_i^n) = \frac{1}{\sqrt{2\pi\langle(x^n)^2\rangle}} \exp\left[-\frac{(x_i^n)^2}{2\langle(x^n)^2\rangle}\right],$$

$$\text{with } \langle(x^n)^2\rangle = \frac{1}{N^2}(NP) = \frac{P}{N}.$$

$$\text{The parameter } \alpha = \frac{P}{N}$$

plays an important role in the theory. The probability of the state $\sigma_i = \xi_i^1$ destabilizing is the probability of the noise term x_i^n overcoming the signal term x_i^s . It is given by

$$\begin{aligned} P[x_i^s + x_i^n < 0] &= P[x_i^n < -x_i^s] = \int_{-\infty}^{-x_i^s} dx P(x) = \frac{1}{2} - \int_0^{x_i^s} dx P(x) \\ &= \frac{1}{2} \left\{ 1 - \operatorname{erf}\left(x^s [2\langle(x^n)^2\rangle]^{-1/2}\right) \right\}, \end{aligned}$$

where the definition of the error function, given by Eq. (2.13), has been used. The probability P^* that *all* states are stable is

$$P^* = \left(1 - P[x_i^n < -x_i^s]\right)^N.$$

We assume that we want a *perfect retrieval* with probability $P^* = 0.5$. Since P^* is of the order of 1, $P[x_i^n < -x_i^s]$ is of the order of 0 and the erf

function is of the order of 1. The argument of the error function must therefore be large and the following asymptotic expansion can be used:

$$\operatorname{erf}(x) \simeq 1 - \frac{1}{x\sqrt{\pi}} \exp(-x^2).$$

One has

$$\frac{1}{2} = \left\{ \frac{1}{2} \left[1 + \operatorname{erf}\left(\frac{x^s}{\sqrt{2\alpha}}\right) \right] \right\}^N \simeq \left\{ 1 - \frac{1}{x^s} \sqrt{\frac{\alpha}{2\pi}} \exp\left(-\frac{(x^s)^2}{2\alpha}\right) \right\}^N.$$

Taking the logarithm of the two sides of this equation and using the expansion $\log(1-x) \simeq -x$ for small x s gives an implicit equation for α

$$\sqrt{\frac{\alpha}{2\pi}} \exp\left[-\frac{(x^s)^2}{2\alpha}\right] = \frac{x^s}{N} \log 2.$$

Taking again the logarithm of both members and neglecting terms of lower orders leads to

$$\alpha \simeq \frac{(1 - 2R/N)^2}{2 \log(N)}.$$

The limit storage capacity $\alpha_c = P_c/N$ is obtained when the radius R shrinks to zero. Moreover it is necessary to take into account a fundamental degeneracy of the network, namely:

If a state $I = \{\sigma_i\}$ is stable, then the state $I' = \{-\sigma_i\}$ is stable.

Proof. — If I is stable all stability parameters x_i^μ are positive

$$x_i^\mu(I) = \sum_j J_{ij} \sigma_i(I) \sigma_j(I) > 0.$$

$$\begin{aligned} \text{Since } \sum_j J_{ij} \sigma_i(I) \sigma_j(I) &= \sum_j J_{ij} (-\sigma_i(I)) (-\sigma_j(I)) \\ &= \sum_j J_{ij} \sigma_i(I') \sigma_j(I') = x_i^\mu(I'), \end{aligned}$$

the stability parameters $x_i^\mu(I')$ are positive and the state I' is stable. It must be emphasized that this result is general. It is a property of the neural networks being connected through quadratic interactions. In particular it remains valid even when the interactions J_{ij} are not symmetrical.

Finally, the maximum number P_c of independent random patterns that a Hebbian network is able to store and to retrieve without any error is

$$P_c = \frac{N}{4 \log(N)}. \quad (4.11)$$

4.1.5 A study of metastable spurious states

a) A geometrical approach

The memorized patterns are stable states of the dynamics of neural networks with synaptic efficacies given by the Hopfield learning rule, at least in the limit of small numbers P of patterns ($P \ll N$). But even in this case there exist other unwanted states, called spurious states, which also are minima of the energy function. Spurious states do exist even though the memorized patterns are orthogonal.

To understand the source of spurious states it is instructive to introduce a geometrical description of the energy function.

We first remark that the iso-energetic surfaces,

$$H(I) = -\frac{1}{2}N \sum_{\mu} (M^{\mu}(I))^2 = C^t,$$

are spheres in the P -dimensional space spanned by the vectors \tilde{M} ; the larger the radius the lower the energy. Moreover \tilde{M} is constrained to stay within a convex polyhedron, which we shall shortly make more precise, and therefore the vertices of the polyhedron represent stable states, most of them being actually metastable states (see Fig. 4.2).

To show how the constraints arise we consider the memorization of $P = 5$ patterns and we look at the scalar product:

$$I \cdot (I^1 + I^2 + I^3 + I^4 + I^5) = I \cdot \Sigma^5.$$

We assume that the patterns are orthogonal, that N is large and that $\alpha = P/N$ is small. The N components of Σ^5 can take one of the following values: $-5, -3, -1, +1, +3, +5$; and the occurrence of each value v is given by the binomial expansion coefficients $\binom{5}{v}$:

$$\begin{array}{rcl} \Sigma^5 & = & \left(\underbrace{\dots 5 \dots}_{\frac{1}{32}N}, \underbrace{\dots 3 \dots}_{\frac{5}{32}N}, \underbrace{\dots 1 \dots}_{\frac{10}{32}N}, \underbrace{\dots -1 \dots}_{\frac{10}{32}N}, \underbrace{\dots -3 \dots}_{\frac{5}{32}N}, \underbrace{\dots -5 \dots}_{\frac{1}{32}N} \right). \\ \text{Number of terms} & & \end{array}$$

The largest possible value of the scalar product is

$$\max(I \cdot \Sigma^5) = \frac{1}{32}N(5 \times 1 + 3 \times 5 + 1 \times 10 + 1 \times 10 + 3 \times 5 + 5 \times 1) = \frac{15}{8}N.$$

Therefore the components of M in the P -space must obey the inequality

$$M^1 + M^2 + M^3 + M^4 + M^5 \leq \frac{15}{8}, \quad \text{a 5-plane.}$$

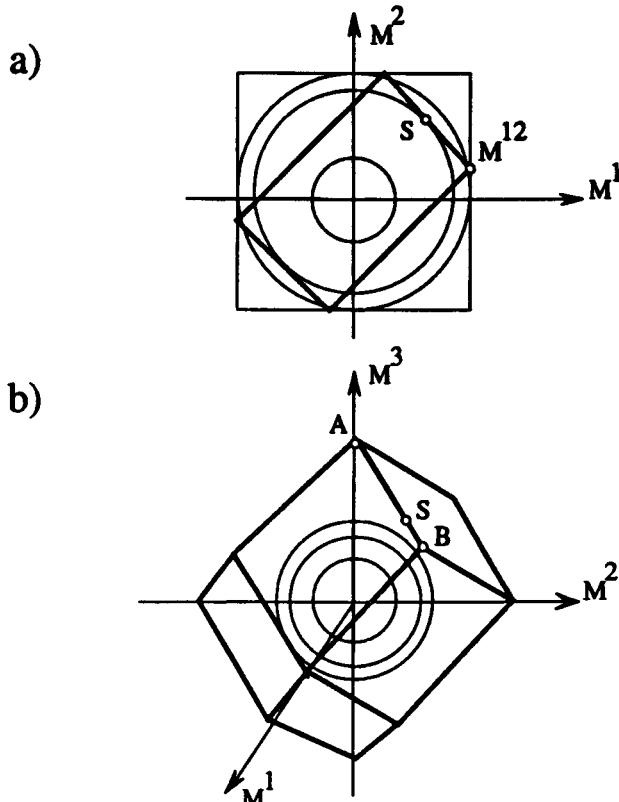


Figure 4.2. The space of overlaps is constrained to convex polyhedrons. The stable states are points of largest modules.
 a) Two non-orthogonal patterns are stored (M^{12} is a stable point)
 b) Three orthogonal patterns are stored. Point A is a ground state. It represents a memorized pattern. Points like B are spurious states. Points S are saddle points.

The same argument yields

$$M^1 + M^2 + M^3 + M^4 \leq \frac{3}{2}, \quad \text{a 4-plane;}$$

$$M^1 + M^2 + M^3 \leq \frac{3}{2}, \quad \text{a 2-plane;}$$

$$M^1 + M^2 \leq 1, \quad \text{a 2-plane.}$$

Replacing one or several M s by $-M$ in the preceding formulae yields a whole set of constraints which delimit a convex polyhedron of legitimate vectors \tilde{M} .

The intersection of five 4-planes is a vertex A of coordinates,

$$M_A^1 = M_A^2 = M_A^3 = M_A^4 = M_A^5 = \frac{1}{4} \times \frac{3}{2} = \frac{3}{8}.$$

This point is on the 5-plane since $\sum_{k=1}^5 M_A^k = 5 \times \frac{3}{8} = \frac{15}{8}$ and therefore this is a stable state.

Similarly, the coordinates of the point B of intersection of four 3-planes are for example

$$M_B^1 = M_B^2 = M_B^3 = M_B^4 = \frac{1}{3} \times \frac{3}{2} = \frac{1}{2} \text{ and } M_B^5 = 0.$$

This point lies outside the polyhedron, since

$$\sum_{k=1}^5 M_B^k = 4 \times \frac{1}{2} = 2 > \frac{15}{8},$$

which does not fulfil the 4-plane condition. A vector $\tilde{M} = \{M, M, M, M, 0\}$ necessarily cuts a 4-plane at a point C which is not a vertex of the polyhedron and therefore it is an unstable state.

The arguments can be made more general: a q -plane is determined by looking at the maximum value of the scalar product $I \cdot \Sigma^q$. It is given by

$$\max(I \cdot \Sigma^q) = \frac{N}{2^q} \left\{ \binom{q}{0} |q| + \binom{q}{1} |q-2| + \binom{q}{2} |q-4| + \dots \right\}$$

and therefore

$$M^1 + M^2 + M^3 + \dots + M^q \leq \frac{1}{2^q} \sum_{k=0}^q \left\{ \binom{q}{k} |q-2k| \right\}. \quad (4.12)$$

The calculation of the sum has been carried out by Amit *et al.* (see below). These authors find

$$\frac{q}{2^q} \binom{q}{\frac{1}{2}q} \quad \text{for even } q \text{ s} \quad \text{and} \quad \frac{q}{2^{q-1}} \binom{q-1}{\frac{1}{2}[q-1]} \quad \text{for odd } q \text{ s}.$$

The coordinates of the point A of intersection of $q(q-1)$ planes, with *odd* qs ,

$$M^1 + M^2 + \dots + M^{q-1} \leq \frac{q-1}{2^{q-1}} \binom{q-1}{\frac{1}{2}[q-1]}$$

is given by $M_A^1 = M_A^2 = \dots = M_A^q = \frac{1}{2^{q-1}} \binom{q-1}{\frac{1}{2}[q-1]}.$ (4.13)

This vertex is on the q -plane, since

$$\sum_{k=1}^q M_A^k = \frac{q}{2^{q-1}} \binom{q-1}{\frac{1}{2}[q-1]}$$

and it is a stable state.

On the other hand, the point of intersection B of $q(q-1)$ planes with *even* qs ,

$$M^1 + M^2 + \dots + M^{q-1} \leq \frac{q-1}{2^{q-2}} \binom{q-2}{\frac{1}{2}[q-2]},$$

is given by $M_B^1 = M_B^2 = \dots = M_B^q = \frac{1}{2^{q-2}} \binom{q-2}{\frac{1}{2}[q-2]}.$

The point is outside the polyhedron, since

$$\sum_{k=1}^q M_B^q = \frac{q}{2^{q-2}} \binom{q-2}{\frac{1}{2}[q-2]},$$

which does not fulfill the q -plane condition and its projection on the polyhedron is unstable.

It is worth noting that the neuronal state corresponding to a stable vertex with q non-zero coordinates given by Eq. (4.13) is

$$I^A = \{\xi_i^A\}, \quad \text{with } \xi_i^A = \text{sign}(\xi_i^1 + \xi_i^2 + \dots + \xi_i^q) \quad \text{and } q \text{ odd.}$$

Let us compute the scalar product $I^A \cdot I^1$, for example. The argument of the sign function is written as

$$\xi_i^1 + \sum_{k=2}^q \xi_i^k.$$

The sum can be either of the sign of ξ_i^1 , then $\xi_i^A = \xi_i^1$, or of the opposite sign, then $\xi_i^A = -\xi_i^1$, or it can be zero, and $\xi_i^A = \xi_i^1$. In the scalar product the first two contributions cancel each other and the last one is the one which matters. The relative number of terms with zero sums is

$$\frac{1}{2^{q-1}} \binom{q-1}{\frac{1}{2}[q-1]},$$

and therefore one finds $M_A^1 = \frac{1}{2^{q-1}} \binom{q-1}{\frac{1}{2}[q-1]}$,

which is indeed the value we found for the coordinates of the vertex A .

A more careful analysis takes the patterns correlations $M^{\mu\mu'}$ into account. For example, the two first series of conditions become

$$M^\mu + M^{\mu'} \leq 1 + M^{\mu\mu'}$$

and $M^\mu + M^{\mu'} + M^{\mu''} \leq \frac{1}{2}(3 + M^{\mu\mu'} + M^{\mu'\mu''} + M^{\mu''\mu})$.

Non-zero correlations distort the polyhedrons, bring about more spurious states and eventually destabilize the memorized states (see Fig. 4.2). We do not want to go further in that direction here. It is worth noting that, for small P s, the number of spurious states increases exponentially with the number of patterns at least as 4^P . The number of symmetrical vertices is

$$\binom{P}{P-1} + \binom{P}{P-3} + \dots = \sum_{k=0}^{(P-1)/2} \binom{P}{2k} = \frac{1}{2} \sum_{k=0}^P \binom{P}{k} = \frac{1}{2} 2^P$$

for the first quadrant. There are 2^P quadrants and the number of symmetrical vertices is $\frac{1}{2} 4^P$.

b) *An analytical derivation*

The following calculations have been carried out by Amit, Gutfreund and Sompolinsky. These authors suppose that the patterns are orthogonal and that the metastable states are symmetrical:

$$\tilde{M} = M \times \left\{ \underbrace{1, 1, 1, \dots}_q, \underbrace{0, 0, 0, \dots}_{P-q} \right\}.$$

Their energies are given by

$$E_q = \frac{2H_q}{N} = -qM^2.$$

The problem is to find M . From the equilibrium conditions one has

$$\sigma_i = \text{sign}(h_i) = \text{sign}\left(\sum_{\mu'=1}^P M^{\mu'} \xi_i^{\mu'}\right)$$

and

$$M^\mu = \frac{1}{N} \sum_i \xi_i^\mu \text{sign}\left(\sum_{\mu'=1}^P M^{\mu'} \xi_i^{\mu'}\right).$$

Summing over all non-zero components of \tilde{M} , one obtains:

$$\sum_{\mu=1}^q M^\mu = q \times M = \frac{1}{N} \sum_i \left\{ \left(\sum_{\mu=1}^q \xi_i^\mu \right) \text{sign}\left(M \sum_{\mu'=1}^q \xi_i^{\mu'}\right) \right\}.$$

The solution of this implicit equation in M depends on the specific *realization* of patterns I^μ that is achieved in the network. Since we are interested in typical properties of networks, however, averages over realizations have to be carried out. An average \bar{O} over the realizations of memorized patterns is defined by

$$\bar{O} = \sum_{\{\xi_i^\mu\}} O(\{\xi_i^\mu\}) P(\{\xi_i^\mu\}),$$

where $P(\{\xi_i^\mu\})$ is, for example, the distribution determined by Eq. (4.8). Letting

$$z_q^i = \sum_{\mu=1}^q \xi_i^\mu$$

and using $M > 0$, the solution is given by

$$M = \frac{1}{q} \overline{z_q \text{sign}(z_q)} = \frac{\overline{|z_q|}}{q}.$$

Then, if k is the number of components with $\xi_i^\mu = +1$, one has

$$z_q = 2k - q, \quad \text{with probability } \frac{1}{2^q} \binom{q}{k}$$

and therefore $\overline{|z_q|} = \frac{1}{2^q} \sum_{k=0}^q \binom{q}{k} |2k - q|$, which is Eq. (4.12). Using $\bar{z}_q = 0$ and the identity

$$\text{sign}(z) = \frac{\text{p.p.}}{i\pi} \int_{-\infty}^{+\infty} \frac{d\theta}{\theta} e^{iz\theta},$$

where p.p. means the principal part, one can write

$$\begin{aligned} \overline{|z_q|} &= \overline{z_q(1 + \text{sign}(z_q))} \\ &= \frac{1}{2^q} \sum_{k=0}^q \binom{q}{k} (2k - q) \left(1 + \frac{\text{p.p.}}{i\pi} \int_{-\infty}^{+\infty} \frac{d\theta}{\theta} e^{i\theta(2k-q)} \right) \\ &= \frac{1}{\pi 2^q} \int_{-\infty}^{+\infty} \frac{d\theta}{\theta} \left(-\frac{d}{d\theta} \left[\sum_{k=0}^q \binom{q}{k} e^{i\theta(2k-q)} \right] \right) \\ &= \frac{1}{\pi 2^q} \int_{-\infty}^{+\infty} \frac{d\theta}{\theta} \left(\frac{d}{d\theta} e^{-iq\theta} [1 + e^{2i\theta}]^q \right), \end{aligned}$$

which yields

$$\overline{|z_q|} = \frac{1}{\pi 2^q} \int_{-\infty}^{+\infty} \frac{d\theta}{\theta} \frac{d}{d\theta} (2 \cos(\theta))^q = \frac{q}{\pi} \int_{-\infty}^{+\infty} \frac{d\theta}{\theta} \sin(\theta) \cos^{q-1}(\theta).$$

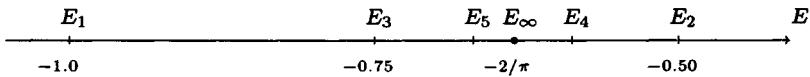
This is a classical integral, the value of which is $\frac{q}{2^q} \binom{q}{\frac{1}{2}q}$ for even qs and $\frac{q}{2^{q-1}} \binom{q-1}{\frac{1}{2}[q-1]}$ for odd qs . Then for even qs one finds

$$M = \frac{1}{2^q} \binom{q}{\frac{1}{2}q} \quad \text{and} \quad E_q = \frac{2H_q}{N} = -\frac{q}{2^{2q}} \binom{q}{\frac{1}{2}q}^2,$$

and for odd qs

$$M = \frac{1}{2^{q-1}} \binom{q-1}{\frac{1}{2}[q-1]} \quad \text{and} \quad E_q = \frac{2H_q}{N} = -\frac{q}{2^{2q-2}} \binom{q-1}{\frac{1}{2}[q-1]}^2.$$

These results provide a hierarchy of metastable levels:



The odd combinations of memorized patterns are stable spurious states, whereas the even combinations are in fact unstable states. The two series converge towards the value $-2/\pi$, which is the ground state energy of a spin glass system with a Gaussian distribution of connections strengths.

Actually, other less symmetrical metastable states such as

$$\tilde{M} = \left\{ \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{4}, \frac{1}{4}, \dots \right\}$$

with $2H/N = -0.688$ also exist.

4.2 Stochastic Hebbian neural networks in the limit of finite numbers of memorized patterns

4.2.1 A reminder of statistical mechanics

Let us summarize the main characteristics of the symmetrically connected Hebbian model:

- A symmetrically connected Hebbian network is defined as a neural network with efficacies given by

$$J_{ij} = \frac{1}{N} \sum_{\mu=1}^P \xi_i^\mu \xi_j^\mu, \quad J_{ii} = 0, \quad J_{i0} = \theta_i = 0,$$

with $i = 1, 2, \dots, N$ and $\mu = 1, 2, \dots, P$.

Since the interactions are symmetrical, $J_{ij} = J_{ji}$, it is possible to define an energy function $H(I)$:

$$H(I) = - \sum_{\langle ij \rangle} J_{ij} \sigma_i(I) \sigma_j(I), \quad (4.14)$$

where the sum is over all pairs $\langle ij \rangle$ of units. When the state dynamics is deterministic (noiseless networks) and asynchronous (as in Glauber dynamics), the energy is a decreasing function of time and the limit behaviors are fixed points.

- The memorized patterns are stable states of the dynamics, at least as long as they are not too numerous, that is to say when $P \ll N$. However, there exist a great many other metastable states (called spurious states) even when the memorized patterns are orthogonal.

This makes the energy landscape in the phase space of the network very tortuous. To gain some information on this landscape it is interesting to build an energy histogram of metastable states: one starts from a random pattern. The asynchronous deterministic dynamics (for $\beta^{-1} = 0$) make the state of the system evolve till some fixed point is obtained. The energy of the state is computed and a point is added to the histogram accordingly. Figure 4.3 shows the energy distributions found in an $N = 200$ unit network for increasing P s. The histograms under the base line refer to the memorized patterns whereas those above the base line include all stable and metastable states. This experiment shows that the energy distribution differentiates two populations of states: the low-energy band is associated with the memorized patterns and the high-energy band is associated with the spurious states. The two parts start merging when the number of patterns is

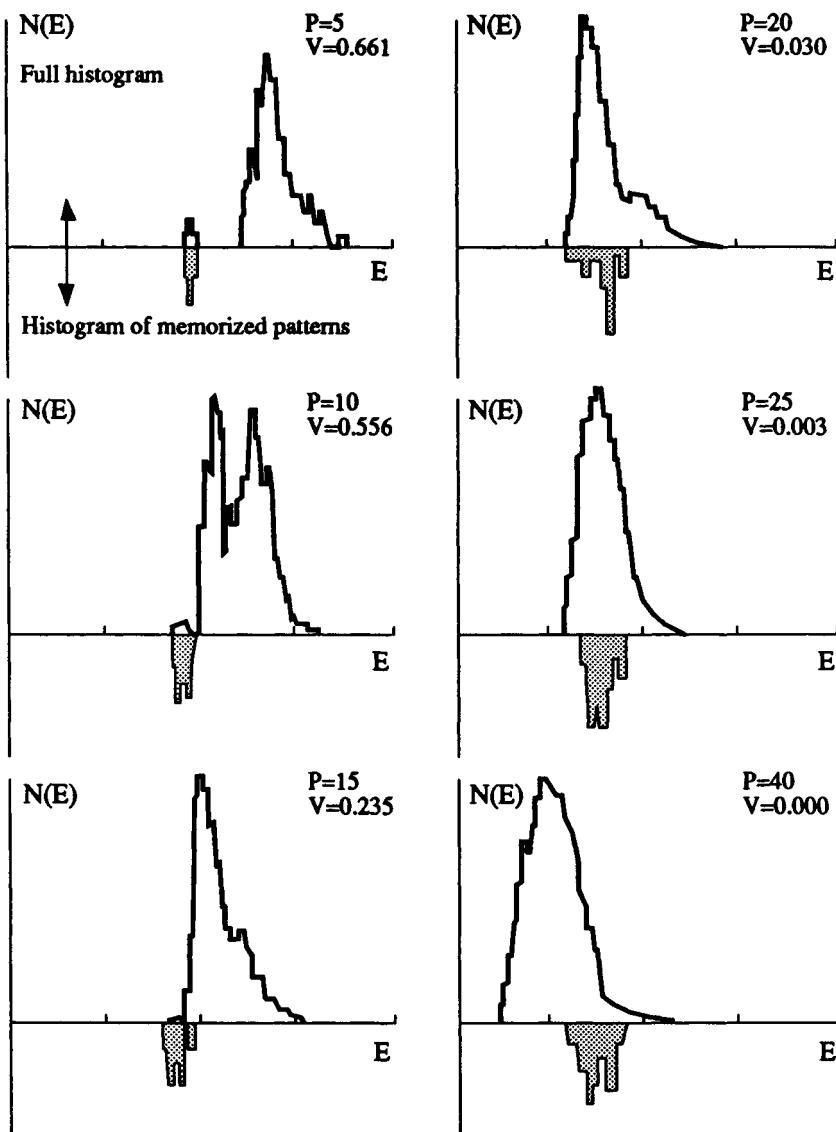


Figure 4.3. Energy histograms of a 200-unit network whose interactions are given by a Hebbian rule. The histograms are built by letting the network go into a fixed point, starting from a random state. P is the number of stored configurations. V is the volume of phase space which is associated with the P memorized patterns. One observes that this volume shrinks to zero for $P \simeq 25$. An accurate theoretical prediction is given in section 4.3. $B = 0$ in the simulations.

about 10 ($\alpha \simeq 0.05$) and the volume occupied by the basins of attractions of the memorized patterns shrinks to zero for a number of patterns of about 25 ($\alpha \simeq 0.125$). The theory of Amit *et al.*, which is explained in detail in section 4.4, accounts for these results.

- Let us introduce noise. Henceforth it is assumed that the neuronal response is in the form of a tanh function. Then we know that the steady distribution of states is given by the Boltzmann statistics,

$$\rho^*(I) = \frac{1}{Z} \exp[-\beta H(I)],$$

where the partition function Z is given by

$$Z = \sum_{I=\{\sigma_i\}} \exp[-\beta H(I)].$$

The central theorem of statistical mechanics, which we do not prove here, is that the free energy $F(\beta)$ defined by

$$F(\beta) = -\beta^{-1} \log(Z)$$

is minimal when the system is steady.

Remark

F may be written as $F = U - \beta^{-1}S$ where the ‘internal energy’ U is the thermal average of H :

$$U = \sum_I \rho^*(I) H(I)$$

and the ‘entropy’ S , a positive quantity, is the thermal average of $-\log(\rho^*)$:

$$S = - \sum_I \rho^*(I) \log(\rho^*(I)).$$

To prove the statement one directly computes F :

$$\begin{aligned} U - \beta^{-1}S &= \frac{1}{Z} \sum_I \exp(-\beta H(I)) \left[H(I) + \beta^{-1}(-\beta H(I) - \log(Z)) \right] \\ &= -\frac{1}{Z} \beta^{-1} \sum_I \exp[-\beta H(I)] \log(Z) = -\beta^{-1} \log(Z) = F. \end{aligned}$$

Let us remember that one is interested in quantities averaged over ensembles of equivalent systems. One calls *order parameters* O combinations of dynamical variables that do not average to zero, $\langle O \rangle \neq 0$, for any value of control parameters and in the thermodynamical limit (when the size N of the system increases to infinity). The game of statistical

mechanics is to detect order parameters, to carry out the computation of the partition function Z and that of the free energy F for *fixed sets of order parameters*

$$F(\beta, \langle O^1 \rangle, \langle O^2 \rangle, \dots),$$

and, finally, to minimize the free energy by solving the following set of equations:

$$\frac{\partial F}{\partial \langle O^1 \rangle} = 0, \dots$$

- It is not always easy to determine the nature of order parameters. Order parameters can simply be guessed and it is verified at the end of the calculation that they do not average to zero. There also exist techniques, such as the Gaussian transform, which, when they can be used, give clues regarding the nature of order parameters. We shall use this technique in the present section. As far as Hebbian models are concerned the overlaps M^μ are valid candidates as order parameters. Let us assume that they are the sole possible order parameters of Hebbian networks. Then the free energy $F = F(\beta, \{M^\mu\} = \widetilde{M})$ determines a landscape in the space of overlaps (the space spanned by the vectors \widetilde{M}). In the limit of zero noise the landscape associated with the free energy $F(\beta^{-1} = 0, \widetilde{M})$ is identical to the landscape determined by the energy $H(\widetilde{M})$ with all its complexities. To prove this statement it is enough to note that in the limit of zero noise the partition function selects the state of lowest energy *for a given* \widetilde{M} . This energy is $H(\widetilde{M})$. Increasing noise makes the landscape of free energy smoother and smoother. Shallow valleys, such as those associated with spurious states, are erased first. As noise still increases, erasing goes on till only one valley is left and every specific property of the network is lost.

- It is important to note that the above description rests on the assumption that the set of order parameters $\{M^\mu\}$ is enough to fully describe the free energy F . This hypothesis is not always valid and we shall see that when the number P of patterns is of the order of N , the number of neurons, other order parameters such as the Edwards Anderson order parameter (noted Q) and the Amit Gutfreund Sompolinsky parameter (noted R) also come into play.

- Finally, neural networks are random systems in the sense that their parameters are determined by the set of random patterns one wants to memorize. The calculations of order parameters are carried out for one realization of interactions (for one set of memorized patterns). One says that disorder is quenched. In actual fact we are looking for properties that are independent of particular realizations of memorized patterns. The calculations then involve two types of averages, thermal averages

and averages over realizations. These various sorts of averages have already been introduced in section 3.3.1 (see Fig. 3.5). The double average is written as $\overline{\langle O \rangle}$ where the bracket is for the thermal (noise induced) average and the upper bar for an average over realizations. How the two averages must be combined is not a trivial problem, which we shall discuss in several places below.

4.2.2 The calculation of order parameters

One strives to compute the partition function

$$Z = \sum_{\{\sigma_i = \pm 1\}} \exp[-\beta H(\{\sigma_i\})],$$

where the energy is given by Eq. (4.14). Z is therefore given by

$$\begin{aligned} Z &= \sum_{\{\sigma_i\}} \exp \left[+\frac{\beta}{2N} \sum_{\mu} \left(\sum_i \xi_i^{\mu} \sigma_i \right)^2 \right] \\ &= \sum_{\{\sigma_i\}} \prod_{\mu=1}^P \exp \left[+\frac{\beta}{2N} \left(\sum_i \xi_i^{\mu} \sigma_i \right)^2 \right]. \end{aligned} \quad (4.15)$$

One uses a trick, the Gaussian transform, whose interest is to linearize the exponent. The transform rests upon the formula

$$\exp(+ax^2) = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{+\infty} dy \exp[-(y^2 - 2xy\sqrt{a})].$$

The partition function becomes

$$Z = \sum_{\{\sigma_i\}} \prod_{\mu=1}^P \frac{1}{\sqrt{\pi}} \int_{-\infty}^{+\infty} dy^{\mu} \exp \left[-y^{\mu 2} + \frac{2\beta^{1/2}}{(2N)^{1/2}} y^{\mu} \left(\sum_i \xi_i^{\mu} \sigma_i \right) \right]. \quad (4.16)$$

The change of variables,

$$M^{\mu} = \left(\frac{2}{N\beta} \right)^{1/2} y^{\mu},$$

introduces quantities M^{μ} which will be soon identified with the overlaps. Then

$$\begin{aligned} Z &= \sum_{\{\sigma_i\}} \left(\frac{N\beta}{2\pi} \right)^{P/2} \prod_{\mu=1}^P \int_{-\infty}^{+\infty} dM^{\mu} \exp \left[-\frac{1}{2} N\beta M^{\mu 2} + \beta M^{\mu} \left(\sum_i \xi_i^{\mu} \sigma_i \right) \right] \\ &= \left(\frac{N\beta}{2\pi} \right)^{P/2} \sum_{\{\sigma_i\}} \int \mathcal{D}\widetilde{M} \exp \left[-\frac{1}{2} N\beta \widetilde{M}^2 \right] \prod_{\mu=1}^P \exp \left[+\beta M^{\mu} \left(\sum_i \xi_i^{\mu} \sigma_i \right) \right]. \end{aligned}$$

It is now possible to carry out the summations over the state variables σ_i . One notes that this amounts to performing the summation over all possible states I of the network *with fixed M* , which therefore becomes a good candidate for a (vectorial) order parameter:

$$\begin{aligned} Z &= \left(\frac{N\beta}{2\pi}\right)^{P/2} \int \mathcal{D}\tilde{M} \exp\left[-\frac{1}{2}N\beta\tilde{M}^2\right] \prod_{i=1}^N 2\cosh(\beta\tilde{\xi}_i \cdot \tilde{M}) \\ &= \left(\frac{N\beta}{2\pi}\right)^{P/2} \int \mathcal{D}\tilde{M} \exp\left[-\frac{1}{2}N\beta\tilde{M}^2 + \sum_i \log 2\cosh(\beta\tilde{\xi}_i \cdot \tilde{M})\right], \end{aligned}$$

where $\tilde{\xi}_i = \{\xi_i^1, \xi_i^2, \dots, \xi_i^P\}$.

In the steepest descent approximation the value of the integral is dominated by the order parameters which minimize the exponent.[†] Therefore one has

$$\frac{F}{N} = -\frac{1}{N\beta} \log(Z) = \frac{1}{2}\tilde{M}^2 - \frac{1}{N\beta} \sum_i \log(2\cosh(\beta\tilde{\xi}_i \cdot \tilde{M})) \quad (4.17)$$

and the order parameters M^μ are given by

$$\frac{\partial F}{\partial M^\mu} = 0 \quad \text{or} \quad M^\mu = \frac{1}{N} \sum_i \xi_i^\mu \tanh(\beta\tilde{\xi}_i \cdot \tilde{M}), \quad (4.18)$$

which yields a set of equations for a given realization $\{\xi_i^\mu\}$ of the set of memorized patterns.

To understand the nature of the dynamical variables, we look for the values M^μ which minimize the exponent *before* the summation over the states variables σ_i has been carried out:

$$\frac{d}{dM^\mu} \left[\frac{1}{2}N(M^\mu)^2 - M^\mu \sum_i \xi_i^\mu \sigma_i \right] = 0.$$

We find $M^\mu = N^{-1} \sum_i \xi_i^\mu \sigma_i$, which is the definition of overlaps. After thermal averaging has been performed one finds

$$M^\mu = \langle M^\mu \rangle = \frac{1}{N} \sum_i \xi_i^\mu \langle \sigma_i \rangle.$$

[†] An introduction to the steepest descent approximation is given at the end of this section.

This formula justifies the introduction of M^μ 's as thermally averaged overlaps. Equations (4.18) can be given a vectorial form:

$$\widetilde{M} = \frac{1}{N} \sum_{i=1}^N \tilde{\xi}_i \tanh(\beta \tilde{\xi}_i \cdot \widetilde{M}), \quad (4.19)$$

which shows that the vector \widetilde{M} is a weighted sum of the N vectors $\tilde{\xi}_i$ representing the set of memorized patterns. The vector \widetilde{M} can be written as $\widetilde{M} = M\tilde{u}$, where \tilde{u} is a unit vector in the P -space. Further still, it is necessary to have some idea regarding the direction of \tilde{u} . According to the discussion on spurious states that we developed in the last section, the metastable states lie in directions of high symmetry. This result suggests that solutions of the type

$$\tilde{u}^q \sqrt{q} = \left\{ \underbrace{1, 1, 1, 1, 1}_{q}, \underbrace{0, 0, 0, 0, 0}_{P-q} \right\}$$

could be good candidates as order parameters (Amit *et al.*). This assumption does not hold as soon as the number of patterns becomes of the order of the number of units. The energy landscape then becomes so disordered that the metastable states loose their symmetry properties. The analysis we are going to develop will be valid no more and a new approach will be needed. It is explained in section 4.3.

From Eq. (4.19), one has:

$$\widetilde{M} \cdot \tilde{u}^q = M = \frac{1}{N} \sum_i \tilde{\xi}_i \cdot \tilde{u}^q \tanh(\beta M(\tilde{\xi}_i \cdot \tilde{u}^q)). \quad (4.20)$$

The quantity

$$z_i^q = \tilde{\xi}_i \cdot \tilde{u}^q = \frac{1}{\sqrt{q}} \sum_{\mu=1}^q \xi_i^\mu$$

is a random variable which varies from site to site. The sum in Eq. (4.20) is an average over the possible values of z^q . It is written as

$$M = \overline{z^q \tanh(\beta z^q M)}. \quad (4.21)$$

It is worth noting that the average in Eq. (4.21) is carried out on *a single realization of patterns*. Theoretically this average would have to be supplemented by an average over many realizations. However, we make here the hypothesis that the system is *self-averaging*, that is to say averaging over a single sample is equivalent to averaging over many samples. It is difficult, in general, to decide whether a system self-averages or not. Computer simulations show that self-averaging is a hypothesis which is fairly well satisfied by Hebbian neural networks.

Consequently, the symbol that is used for averaging over realizations may be kept in Eq. (4.21).

The amplitude M is a measure of the quality of the retrieval of a state $M\tilde{u}^q$. For large enough noises M shrinks to zero in the very same way as magnetism disappears in a ferromagnet when temperature reaches a critical value (the Curie point). The critical point is found by expanding the r.h.s. of Eq. (4.21) to third order:

$$M \simeq \overline{z^q \beta M z^q (1 - \frac{1}{3}(\beta M z^q)^2)} = \beta M - \frac{1}{3} \beta^3 M^3 \overline{(z^q)^4}.$$

z^q is a random variable whose distribution is given by the binomial coefficients $\binom{q}{z^q}$ (a random walk of q steps $\mp 1/\sqrt{q}$). The fourth moment of this distribution is

$$\overline{(z^q)^4} = 3 - \frac{2}{q} \quad \text{and, for } \beta \simeq 1, \quad M^2 \simeq \frac{\beta - 1}{1 - \frac{2}{3}q},$$

which shows that the critical noise is $\beta_c = 1$ for any symmetrical order parameter. The transition to zero amplitude is smooth. It is said that the transition is a *second-order transition*. In actual fact the analysis of the stability shows that, as noise increases, a catastrophic *first-order transition* occurs before noise reaches the critical value for all symmetrical solutions except for those with one component ($q = 1$, which correspond to the memorized states). As β decreases (increasing noise) the various metastable states are destabilized in turn. This analysis, which we owe to Amit *et al.*, is given in the next section.

4.2.3 The stabilities of symmetrical solutions

The stabilities of solutions are determined by the eigenvalues of the Jacobian A of F . One has

$$\begin{aligned} A^{\mu\mu'} &= \frac{\partial^2 F}{\partial M^\mu \partial M^{\mu'}} \\ &= \delta^{\mu\mu'} - \beta \left[\frac{1}{N} \sum_i \xi_i^\mu \xi_i^{\mu'} \left(1 - \tanh^2 \left[\beta \sum_{\mu'} \xi_i^{\mu'} M^{\mu'} \right] \right) \right] \\ &= \delta^{\mu\mu'} - \beta \left[\delta^{\mu\mu'} - \overline{\xi^\mu \xi^{\mu'} \tanh^2(\beta \tilde{\xi} \cdot \widetilde{M})} \right], \end{aligned} \quad (4.22)$$

where the bar is again an average over a distribution of memorized patterns. For random patterns the distribution is given by Eq. (4.8):

$$P(\{\xi_i^\mu\}) = \prod_{\mu,i} \frac{1}{2} (\delta(\xi_i^\mu - 1) + \delta(\xi_i^\mu + 1)).$$

Letting $a^{\parallel} = \overline{(\xi^\mu \tanh(\beta \tilde{\xi} \cdot \widetilde{M}))^2} = \overline{(M^\mu)^2}$

and

$$a^{\perp} = \overline{\xi^\mu \xi^{\mu'} \tanh^2(\beta \tilde{\xi} \cdot \widetilde{M})},$$

the Jacobian matrix takes the following form:

$$A = \begin{pmatrix} A^{\parallel} & A^{\perp} & \cdots & & \\ A^{\perp} & A^{\parallel} & & & \\ \vdots & & \ddots & & \\ \hline & & & A^{\parallel} & 0 \\ 0 & & & 0 & \ddots \\ & & & 0 & A^{\parallel} \end{pmatrix}$$

with $A^{\parallel} = 1 - \beta(1 - a^{\parallel})$, $A^{\perp} = \beta a^{\perp}$.

There are:

- one non-degenerate eigenvalue: $\lambda_1 = A^{\parallel} + (q - 1)A^{\perp}$;
- $(P - q)$ eigenvalues: $\lambda_2 = A^{\parallel}$; and
- $(q - 1)$ eigenvalues: $\lambda_3 = A^{\parallel} - A^{\perp}$.

It can be proved that $A^{\perp} > 0$ (see Amit *et al.*) and therefore λ_3 is the smallest eigenvalue, the one which determines the stabilities of the solutions. The critical noises are given by the relation $\lambda_3 = 0$, that is

$$\begin{aligned} 1 - \frac{1}{\beta} &= \frac{(1 - \xi^\mu \xi^{\mu'}) \tanh^2(\beta \tilde{\xi} \cdot \tilde{M})}{(1 - \xi^\mu \xi^{\mu'}) \tanh^2(\beta M(\beta) z^q)}. \end{aligned} \quad (4.23)$$

The solutions of Eq. (4.23) are computed numerically for $q = 1, 3, 5, \dots$. The solutions for even qs are unstable. One finds:

- $q = 1$: $\beta_1^{-1} = 1.0$. These states $\{1, 0, 0, 0, \dots\}$, called the Mattis states, are the memorized patterns.
- $q = 3$: $\beta_3^{-1} = 0.461$. The states $\{1, 1, 1, 0, 0, \dots\}$ are the most stable spurious states.
- $q = 5$: $\beta_5^{-1} = 0.385$. They are states of the form $\{1, 1, 1, 1, 1, 0, 0, \dots\}$.
- etc.

This analysis shows that the mixtures of an odd number of patterns, except the pure memorized states, are destabilized by the increase of noise. In particular one concludes that for $0.461 < B = \beta^{-1} < 1.0$ all states except the Mattis states are unstable. Noise in this range of values therefore washes out all spurious states and, far from being a hindrance, helps in faithful pattern retrieval.

More details are given in the article by Amit *et al.*

4.2.4 On the method of steepest descent and mean field approximations (also called the saddle point method)

We are interested in the asymptotic expansion of the integral

$$Z(N) = \int_{-\infty}^{+\infty} dM \exp(-N f(M)) \quad (4.24)$$

along the real axis when N goes to infinity. $f(M)$ so behaves that the integrand goes to zero faster than $1/|M|$ for $|M| \rightarrow \infty$. Moreover we assume that f does not display any essential singularities. The integration can also be carried out in the complex plane by choosing any integration contour Γ which makes a closed loop with the real axis: The integral along Γ is the same as the integral computed on the real axis (see Fig. 4.4).

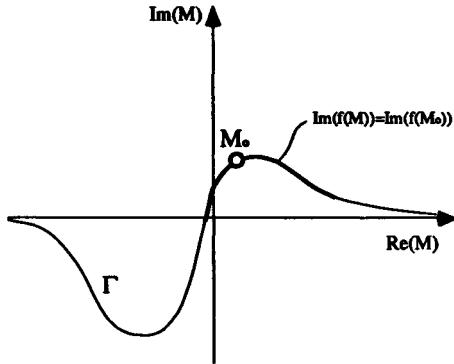


Figure 4.4. The saddle point method. M_0 is the saddle point. Integration can be limited to the bold-lined portion of Γ .

The most convenient contour passes through the point M_0 (the saddle point) determined by

$$\frac{df(M)}{dM} \Big|_{M_0} = 0$$

and is defined by the equation

$$\text{Im}(f(M)) = \text{Im}(f(M_0)). \quad (4.25)$$

Let us expand f in a Taylor series *along the path* Γ up to second order:

$$f(M) = f(M_0) + \frac{1}{2} f''(M_0)(M - M_0)^2.$$

Thanks to Eq. (4.25) the second-order term of the expansion is *real* on Γ and one can write

$$\begin{aligned} Z(N) &= \exp(-N f(M_0)) \int_{\Gamma} dM \exp\left[-\frac{1}{2} N f''(M_0)(M - M_0)^2\right] \\ &= \exp(-N f(M_0)) \frac{1}{\sqrt{N |f''(M_0)|}} \int_{-\infty}^{+\infty} dt \exp\left(-\frac{1}{2} t^2\right), \end{aligned}$$

where t is a curvilinear variable along Γ . Let us now consider a partition function Z :

$$Z = \sum_{\{\sigma_i\}} \exp[-\beta H(\{\sigma_i\})],$$

where the summation is over all states $I = \{\sigma_i\}$ of the system. We introduce a dynamic variable M which we call an order parameter as a certain mapping of states I of the system onto the set of real numbers $M = M(I)$. One can write

$$Z = \sum_M \sum_{\{\sigma_i|M\}} \exp[-\beta H(\{\sigma_i\})],$$

and one assumes that the internal summation yields a formula similar to Eq. (4.24):

$$Z(N) = \int_{-\infty}^{+\infty} dM \exp(-N\beta f(M)).$$

Then

$$Z(N) = \exp(-N\beta f(M_0)) \sqrt{\frac{2\pi}{N\beta |f''(M_0)|}}.$$

The free energy F/N of the system per unit is given by

$$F/N = -\frac{1}{N\beta} \log(Z) = f(M_0) + \frac{1}{2N\beta} \log(N\beta|f''(M_0)|).$$

The last term is negligible in the limit of large N s. We note that it would have been equivalent to define a free energy as

$$\frac{F}{N} = f(M),$$

to take the exponent of Eq. (4.24) as the free energy and to consider that the dynamics of the system strives to minimize this free energy according to the basic theorem of statistical mechanics. This is what we did in Eq. (4.17).

- Let us assume that the variable M of the integral (4.24) is constrained to a condition such as

$$g(M) = R.$$

The integral becomes

$$Z = \int_{-\infty}^{+\infty} dM \exp(-Nf(M)) \delta(g(M) - R) \quad (4.26)$$

and, introducing the integral representation of the δ -function,

$$\delta(x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} dQ \exp(iQx),$$

it is transformed into

$$Z = \int_{-\infty}^{+\infty} dM \frac{1}{2\pi} \int_{-\infty}^{+\infty} dQ \exp \left[-N \left(f(M) + iQ(g(M) - R) \right) \right].$$

Changing Q into iQ , one can write

$$Z = \int_{-\infty}^{+\infty} dM \frac{1}{2\pi i} \int_{-i\infty}^{+i\infty} dQ \exp \left[-N \left(f(M) + Q(g(M) - R) \right) \right],$$

and we apply the saddle point integration to both variables M and Q using the corresponding convenient paths Γ_M and Γ_Q . This yields

$$Z = \exp \left[-Nf(M_0, Q_0) \right],$$

where $f(M, Q) = f(M) + Q(g(M) - R)$ (4.27)

and M_0 and Q_0 are solutions of

$$\frac{\partial f(M, Q)}{\partial M} \Big|_{M_0} = 0, \quad \frac{\partial f(M, Q)}{\partial Q} \Big|_{Q_0} = 0.$$

We recognize the classical Lagrange equations which determine the minimum of a function $f(M)$ given a constraint $g(M) = R$ that M has to obey. Q is the Lagrange multiplier.

It is this technique which will be used later, in section 6.3.2 in particular.

- The method of steepest descent is equivalent to (it yields the same results as) a very popular technique of statistical mechanics, namely the *mean field approximation*.

The mean field approximation rests upon the hypothesis that all correlations between dynamic variables can be neglected. This amounts to assuming that the steady distribution of states factorizes in local distributions:

$$\rho^*(I) = \prod_i \rho_i^*(\sigma_i),$$

with $I = \{\sigma_i\}$. $\rho_i^*(\sigma_i)$ is the steady probability for neuron i to be in the state σ_i and

$$\sum_{\{\sigma_i=\mp 1\}} \rho_i^*(\sigma_i) = 1.$$

The expectation value of the product of two local observables O_i and O_j is given by

$$\begin{aligned} \langle O_i O_j \rangle &= \sum_I \rho^*(I) O_i(I) O_j(I) \\ &= \left[\sum_{\{\sigma_i=\mp 1\}} \rho_i^*(\sigma_i) O_i(\sigma_i) \right] \left[\sum_{\{\sigma_j=\mp 1\}} \rho_j^*(\sigma_j) O_j(\sigma_j) \right] \\ &= \langle O_i \rangle \langle O_j \rangle, \end{aligned}$$

where the normalization of local distributions has been used. It is worth noting that the mean field approximation replaces the average of a function of dynamic variables by the function of its averaged arguments,

$$\begin{aligned} \langle S(h_i) \rangle &= \left\langle S[\langle h_i \rangle + (h_i - \langle h_i \rangle)] \right\rangle \\ &= S[\langle h_i \rangle] + \frac{1}{2} \langle [h_i - \langle h_i \rangle]^2 \rangle S''[\langle h_i \rangle] + \dots = S[\langle h_i \rangle], \end{aligned}$$

since, in particular, the approximation yields $\langle h_i^2 \rangle = \langle h_i \rangle \langle h_i \rangle = \langle h_i \rangle^2$.

To show the equivalence between the steepest descent method and the mean field approximation we compute the average activity of a given neuron i . Since fluctuations are neglected, ($\langle h_i^2 \rangle = \langle h_i \rangle^2$), the steady distribution ρ_i^* is obtained by fixing all other dynamical variables to their expectation values $\langle \sigma_j \rangle$. One has

$$\rho_i^*(\sigma_i) = \frac{1}{Z_i} \exp \left[-\beta \sum_j J_{ij} \sigma_i \langle \sigma_j \rangle \right] = \frac{1}{Z_i} \exp \left[-\beta \sigma_i \langle h_i \rangle \right],$$

where Z_i is the normalization factor:

$$Z_i = \sum_{\sigma_i \in \{+1, -1\}} \exp \left[-\beta \sigma_i \langle h_i \rangle \right] = 2 \cosh(\beta \langle h_i \rangle).$$

For example, the thermally averaged activity of neuron i is given by

$$\langle \sigma_i \rangle = \frac{1}{Z} \sum_{\sigma_i \in \{+1, -1\}} \sigma_i \exp \left[-\beta \sigma_i \langle h_i \rangle \right] = \tanh(\beta \langle h_i \rangle).$$

The local field $\langle h_i \rangle$ is

$$\langle h_i \rangle = \frac{1}{N} \sum_j \sum_\mu \xi_i^\mu \xi_j^\mu \langle \sigma_j \rangle = \sum_\mu \xi_i^\mu M^\mu = \tilde{\xi}_i \cdot \tilde{M}$$

and

$$M^\mu = \frac{1}{N} \sum_i \xi_i^\mu \langle \sigma_i \rangle = \frac{1}{N} \sum_i \xi_i^\mu \tanh(\beta \tilde{\xi}_i \cdot \tilde{M}),$$

which is Eq. (4.18).

The interest of the mean field approximation over the steepest descent technique is that it is more straightforward. The problem is that it does not provide any clue regarding the nature of order parameters and also it is sometimes difficult to estimate how good the approximations are and where improvements could be brought about.

4.2.5 A mean-field theory of the dynamical properties of Hebbian neural networks

We have seen in the last section how the techniques of statistical mechanics may be used to compute the steady properties of symmetrical Hebbian networks whose sizes N are very large with respect to the number P of patterns. Simulations displayed in Figs 4.5 and 4.6 show time evolutions of such systems. To study the dynamical properties it is necessary to appeal to Eq. (3.49) (an exact equation):

$$\frac{d\langle \sigma_i \rangle}{dt} = -\frac{1}{\tau_r} \left[\langle \sigma_i \rangle - \langle S(\beta h_i) \rangle \right].$$

In actual fact the scope of this equation goes beyond the limits of statistical mechanics since $S(h)$ can be *any sigmoidal response function*. For example, $S(h)$ may be an error function if Gaussian synaptic noise

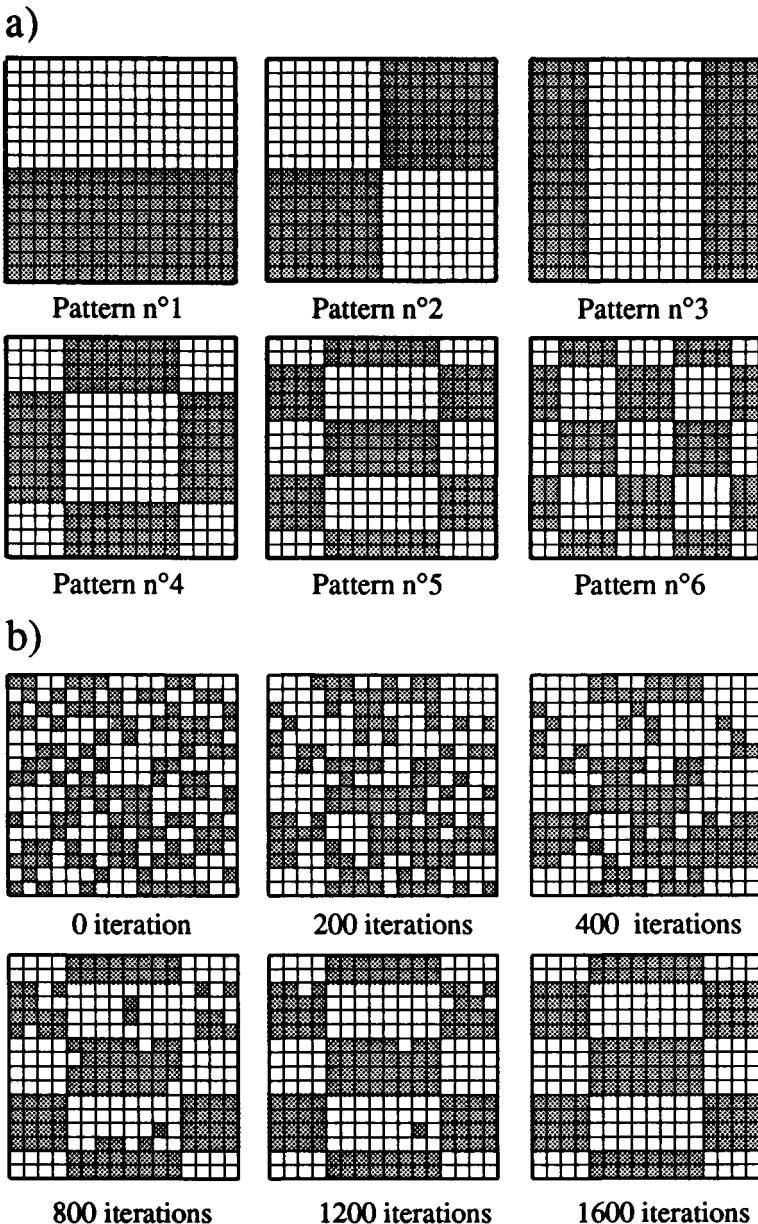


Figure 4.5. Evolution of the neuronal states in a 256-unit neural network. Six orthogonal patterns (displayed in a)) have been stored. The system evolves according to a Glauber dynamics, with a noise parameter $\beta^{-1} = 0.15$. It starts from a random configuration and after about 1000 iterations (four Monte-Carlo steps per neuron) it retrieves pattern n° 5.

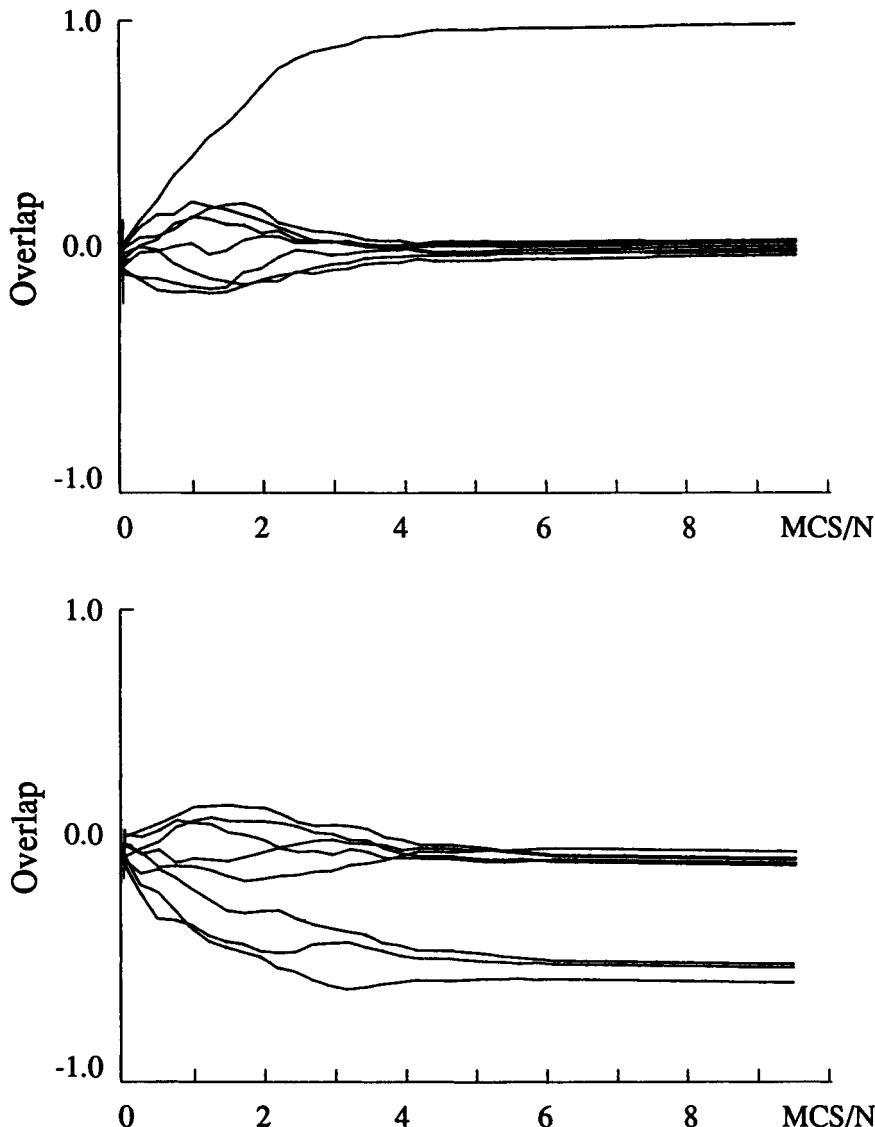


Figure 4.6. An example of spurious states in a 100-neuron network. Eight random patterns have been stored and a zero-noise Glauber dynamics has been used. The upper figure shows that the system retrieves a well-defined pattern, whereas the lower figure is an example of a convergence towards a mixture of three patterns. The theory shows that ternary mixtures are the most stable of the metastable states.

is thought to be more realistic than the thermodynamical noise we have used so far.

Exact solutions of the dynamical equation are difficult to obtain and we must be content with some approximation. The most straightforward one is the mean field approximation, which, as we explained in the note of the preceding section, consists in neglecting the correlations between the dynamical variables. Then

$$\langle S(\beta h(i)) \rangle = S(\beta \langle h_i \rangle).$$

$$\begin{aligned} \text{One has } \frac{d\langle \sigma_i \rangle}{dt} &= -\frac{1}{\tau_r} \left[\langle \sigma_i \rangle - S\left(\beta \sum_{j=0}^N J_{ij} \langle \sigma_j \rangle\right) \right] \\ &= -\frac{1}{\tau_r} \left[\langle \sigma_i \rangle - S\left(\beta \sum_{\mu} \xi_i^{\mu} \frac{1}{N} \left[\sum_j \xi_j^{\mu} \langle \sigma_j \rangle \right]\right) \right] \\ &= -\frac{1}{\tau_r} \left[\langle \sigma_i \rangle - S\left(\beta \sum_{\mu} \xi_i^{\mu} M^{\mu}\right)\right], \end{aligned}$$

$$\begin{aligned} \text{whence } \frac{dM^{\mu}}{dt} &= \frac{1}{N} \sum_i \xi_i^{\mu} \frac{d\langle \sigma_i \rangle}{dt} \\ &= -\frac{1}{\tau_r} \left[M^{\mu} - \frac{1}{N} \sum_i \xi_i^{\mu} S\left(\beta \sum_{\mu'} \xi_i^{\mu'} M^{\mu'}\right) \right] \quad (4.28) \end{aligned}$$

$$\text{and } \frac{d\widetilde{M}}{dt} = -\frac{1}{\tau_r} \left[\widetilde{M} - \overline{\tilde{\xi} S(\beta \tilde{\xi} \cdot \widetilde{M})} \right].$$

When the system is stationary, $d\widetilde{M}/dt = 0$, Eq. (4.19) is recovered. This is no surprise since it has been shown that the mean field approximation is equivalent to the steepest descent method. The interesting point here is that there is no restriction on the shape of $S(h)$.

Let us apply Eq. (4.28) to the dynamics around the critical point. We shall limit our analysis to Mattis states ($q = 1$) and we assume that the synaptic noise is Gaussian. For large noises $\beta \rightarrow 0$ the order parameter M^{μ} vanishes. Using the expansion

$$S(x) = \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right) \simeq (2/\pi)^{1/2} (x - \frac{1}{6}x^3),$$

the equation (4.28) becomes

$$\frac{dM^{\mu}}{dt} = -\frac{1}{\tau_r} \left[M^{\mu} - (2/\pi)^{1/2} \overline{\xi^{\mu} (\beta \tilde{\xi} \cdot \widetilde{M} - \frac{1}{6} \beta^3 (\tilde{\xi} \cdot \widetilde{M})^3)} \right]$$

$$\begin{aligned} &= -\frac{1}{\tau_r} \left[M^\mu - (2/\pi)^{1/2} \sqrt{\beta M^\mu (1 - \frac{1}{6}(\beta M^\mu)^2)} \right] \\ &= -\frac{M^\mu}{\tau_r} \left[1 - \frac{\beta}{\beta_c} (1 - \frac{1}{6}(\beta M^\mu)^2) \right], \end{aligned}$$

with

$$\beta_c^{-1} = \sqrt{\frac{2}{\pi}}.$$

The steady value $M^{\mu*}$ of M^μ is given by the condition $dM^{\mu*}/dt = 0$:

$$(M^{\mu*})^2 = \frac{6(\beta - \beta_c)}{\beta^3}$$

and to first order in $M^\mu - M^{\mu*}$ one finds:

$$\frac{d(M^\mu - M^{\mu*})}{dt} = -\frac{M^\mu - M^{\mu*}}{\tau(\beta)}, \quad (4.29)$$

with

$$\tau(\beta) = \frac{\tau_r}{2(\beta/\beta_c - 1)}.$$

This equation shows that memory is lost for noise $\beta^{-1} > \beta_c^{-1}$. As noise progressively increases, retrieval time becomes longer and longer and the number of errors increases. The relaxation time that is necessary to recall a memorized pattern diverges at critical noise β_c^{-1} as $(\beta - \beta_c)^{-1}$. This effect is known in the theory of critical phenomena as the *critical slowing down*. The number of correctly retrieved bits decreases as $(\beta - \beta_c)^{1/2}$. The simulations displayed in Fig. 4.7 clearly confirm these predictions.

If the mean field approach is appealing because it is easier to understand than other techniques such as the steepest descent, it must be stressed that this easiness hides difficulties which sometimes cannot be ignored for fear of erroneous results. We might also wonder how stability analysis can be performed, since the mean field approximation does not appeal to the idea of a free energy. In fact it is possible to carry out such an analysis.

We write Eqs (4.28) as

$$\frac{dM^\mu}{dt} = -\frac{1}{\tau_r} F_\mu$$

and we compute the following quantities:

$$\begin{aligned} A_{\mu\mu'} &= \frac{\partial F_\mu}{\partial M^{\mu'}} = \delta^{\mu\mu'} - \beta \frac{1}{N} \sum_i \xi_i^\mu \xi_i^{\mu'} S'(\beta \tilde{\xi}_i \cdot \tilde{M}) \\ &= \delta^{\mu\mu'} - \beta \overline{\xi^\mu \xi^{\mu'} S'(\beta \tilde{\xi} \cdot \tilde{M})}. \end{aligned}$$

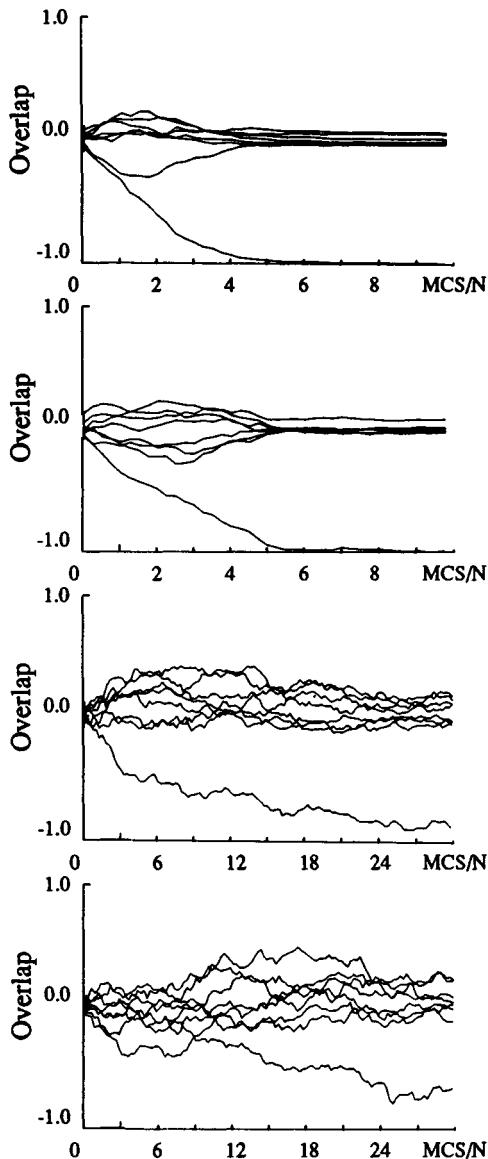


Figure 4.7. The effect of noise on the retrieval dynamics of a 400-unit network. Eight stochastically independent patterns have been stored and the evolution of the relative overlap of the running state with respect to the eight stored configurations have been observed. The dynamics is that of Glauber with a thermodynamical noise. The noise parameters, from top to bottom, are: $\beta^{-1} = 0, 0.50, 0.80$ and 0.95 .

As usual the bar indicates that an averaging over realizations has to be carried out. The expression is symmetrical in the indices μ and μ' . This shows that the F_μ s are forces which are derived from a certain potential F , which, if $S(x) = \tanh(x)$, is the free energy displayed in Eq. (4.17).

The stability of the solutions is determined by the smallest eigenvalue of the Jacobian matrix with elements $A_{\mu\mu'}$. With the more realistic response function $S(x) = \text{erf}(x/\sqrt{2})$ these elements are given by

$$A_{\mu\mu'} = \delta^{\mu\mu'} - \frac{1}{2}\beta \overline{\xi^\mu \xi^{\mu'} \exp -\frac{1}{2}\beta(\tilde{\xi} \cdot \widetilde{M})^2}$$

and the analysis proceeds as in section 4.2.4.

4.3 Storing an infinite number of patterns in stochastic Hebbian networks: the technique of field distributions

4.3.1 Local field distributions in disordered systems

The local field h_i on neuron i can be written as

$$h_i = \sum_j J_{ij} \sigma_j = \sum_\mu \left(\frac{1}{N} \sum_j \xi_j^\mu \sigma_j \right) \xi_i^\mu = \sum_\mu M^\mu \xi_i^\mu = \sum_\mu h_i^\mu.$$

If one of the overlaps, say M^1 , is large and of the order of 1, the other overlaps M^μ ($\mu \neq 1$), assuming that their magnitudes are roughly the same, are necessarily of the order of $1/\sqrt{N}$. We have seen that the pattern M^1 is stable, which means that the partial field h_i^1 overcomes all other partial fields h_i^μ , as long as the number P of patterns remains finite, ($P \simeq O(N^0)$). When P is of the order of N , ($P \simeq \alpha N$), all partial fields h_i^μ add up randomly and eventually destabilize the pattern I^1 . It is then necessary to carry out a double average, a thermal average as usual and an average over different realizations of memorized patterns. When finite numbers of patterns were considered a mere average over the sites of the network for a given realization of memorized patterns was enough. This is no longer true here.

There exist tools in statistical mechanics to tackle the problem of disorder. They have been developed in the context of spin glass theories. One of these techniques is the replica trick which has been introduced by Sherrington and Kirkpatrick to solve the problem of infinite range spin glasses. The replica trick has been applied to neural networks by Amit, Gutfreund and Sompolinsky. In spite of its technicality we think it interesting to explain this theory in detail because it is an achievement in itself and because it can and has been used to solve other aspects of the theory of neural networks. However, the replica trick is a little involved

and we prefer to first introduce a method which is more transparent but less rigorous. (Although it can be argued that the mathematical status of the replica method is not fully established.) We therefore explain the technique of the field-distributions in this section and postpone the explanation of the replica approach to the next.

To carry out the double average one considers N^g realizations of the P memorized patterns. Each realization determines a network Σ^g and its set of interactions $\{J_{ij}^g\}$. As usual in the averaging process, the realizations must obey some constraint C^g . Here the constraint is that *all patterns I^μ are chosen at random except one particular pattern I^1 which must be left unchanged*. On the other hand each system Σ^g is duplicated in N^s equivalent systems Σ^{gs} which undergo different noise histories but whose noise parameters β are the same.

Let $N^g(I)$ be the number of systems Σ^{gs} which are in the same state I for a given realization g of memorized patterns and $N(I)$ the number of systems Σ^{gs} which are in state I for all realizations. The ratio

$$\rho_g^*(I) = \frac{N^g(I)}{N^s}$$

is the thermal distribution probability for the set of systems Σ^g and the ratio

$$\rho^*(I) = \frac{N(I)}{N^s N^g}$$

is the combined distribution for the whole set of systems. One has

$$\rho^*(I) = \frac{1}{N^g} \sum_g \rho_g^*(I).$$

The ensemble average of a local observable O_i is given by

$$\langle \overline{O_i} \rangle = \sum_I \rho^*(I) O_i(I) = \frac{1}{N^g} \sum_g \langle O_i \rangle_g,$$

where $\langle O_i \rangle_g$ is the thermal average of O_i for system Σ^g .

We know that the average of local observables, that of σ_i in particular, is determined by the local field h_i . It is therefore natural to carry out the summation over the $N^s N^g$ systems so as to collect all those giving rise to the same field h_i . Then the summation over the states I is to be replaced by a summation over the local fields:

$$\sum_I \rho^*(I) \cdots \mapsto \sum_{h_i} \sum_{\{\sigma_i\}} \rho_h^*(h_i) w_i(\sigma_i | h_i) \cdots,$$

where $\rho_h^*(h_i)$ is the probability of finding a local field h_i and $w_i(\sigma_i \mid h_i)$ is the probability for i to be in state σ_i given h_i . $\rho_h^*(h_i)$ is determined by the equation

$$\rho_h^*(h_i) = \frac{1}{N^g} \sum_g \sum_I \rho_g^*(I) \delta\left(h_i - \sum_j J_{ij}^g \sigma_j(I)\right).$$

In so far as the realizations are random it seems legitimate to assume that h_i is a sum of random, uncorrelated contributions. Then the distribution ρ_h^* of h_i is a Gaussian distribution whose first and second moments $\langle h_i \rangle$ and $\langle \Delta h_i^2 \rangle$ are given by

$$\langle h_i \rangle = \sum_{h_i} \rho_h^*(h_i) h_i = \frac{1}{N^g} \sum_g \sum_I \rho_g^*(I) h_i^g(I) = \frac{1}{N^g} \sum_g \langle h_i \rangle_g$$

and $\langle \Delta h_i^2 \rangle = \frac{1}{N^g} \sum_g \sum_I \rho_g^*(I) [h_i^g(I) - \langle h_i \rangle]^2,$

with $h_i^g(I) = \sum_j J_{ij}^g \sigma_j(I)$. Owing to constraint C^g , $\langle h_i \rangle$ does not necessarily vanish. At equilibrium the ensemble averaged activity of a unit i is given by

$$\begin{aligned} \langle \sigma_i \rangle &= \sum_{h_i} \rho_h^*(h_i) \sum_{\sigma_i \in \{\pm 1\}} w_i(\sigma_i \mid h_i) \sigma_i \\ &= \sum_{h_i} \rho_h^*(h_i) S(\beta h_i) \\ &= [2\pi \langle \Delta h_i^2 \rangle]^{-1/2} \int dh_i \exp\left[-\frac{(h_i - \langle h_i \rangle)^2}{2\langle \Delta h_i^2 \rangle}\right] S(\beta h_i), \end{aligned}$$

which with the change of variable, $z = \frac{h_i - \langle h_i \rangle}{[\langle \Delta h_i^2 \rangle]^{1/2}}$, becomes

$$\langle \sigma_i \rangle = \frac{1}{\sqrt{2\pi}} \int dz \exp(-\frac{1}{2}z^2) S\left[\beta\left(z \sqrt{\langle \Delta h_i^2 \rangle} + \langle h_i \rangle\right)\right]. \quad (4.30)$$

We apply this formalism to various cases.

4.3.2 The Sherrington Kirkpatrick model of long-range spin glasses

In the limit of very large numbers of memorized patterns the synaptic efficacies

$$J_{ij} = \frac{1}{N} \sum_{\mu=1}^P \xi_i^\mu \xi_j^\mu$$

are sums of random numbers and therefore they are Gaussian variables. The neural networks become similar to long-range spin glasses, a model which received much attention from physicists. Long-range spin glasses are systems of fully connected units interacting through random symmetrical interactions. To describe the behavior of spin glasses it is necessary to introduce a new order parameter Q , the Edwards Anderson parameter. Since for large P s neural networks are bound to behave as long-range spin glasses their description also appeals to Q . It is therefore interesting to start the study of heavily imprinted neural networks by that of spin glasses.

In spin glasses a thermal average and an average over realizations have to be carried out. The constraint \mathcal{C}^g on the average over realizations is that the sets of interactions $\{J_{ij}^g\}$ must have the same first two moments $\bar{J} = J_0$ and $\overline{\Delta J^2} = J_1$. Taking $J_0 \neq 0$ introduces a ‘ferromagnetic bias’ in the interactions, which means that the spins show a tendency to align. The constraint that the pattern I^1 must be left unchanged in all realizations also introduces a (local) bias in the Hopfield model. The interactions of the spin glass are therefore given by

$$J_{ij}^g = \overline{J_{ij}} + \Delta J_{ij}^g = \frac{J_0}{N} + \Delta J_{ij}^g,$$

$$\text{with } \overline{(\Delta J_{ij}^g)^2} = \frac{J_1}{N}.$$

The first moment of the fields distribution is

$$\overline{\langle h_i \rangle} = \frac{1}{N^g} \sum_g \left\langle \sum_j J_{ij}^g \sigma_j \right\rangle_g = J_0 M + \overline{\langle \Delta h_i \rangle} = J_0 M,$$

$$\text{with } M = \frac{1}{N} \sum_j \overline{\langle \sigma_j \rangle}.$$

For the sake of symmetry the field fluctuations

$$\overline{\langle \Delta h_i \rangle} = \frac{1}{N^g} \sum_{g,j} \Delta J_{ij}^g \sigma_j$$

strictly vanish for $J_0 = 0$ and for $J_0 \gg J_1$. It is assumed that they remain negligible in the whole range of J_0 . The second moment is

$$\begin{aligned} \overline{\langle \Delta h_i^2 \rangle} &= \overline{\langle h_i^2 \rangle} - (\overline{\langle h_i \rangle})^2 \\ &= \frac{1}{N^g} \sum_g \sum_I \rho^g(I) \left(\sum_j \Delta J_{ij}^g \sigma_j(I) \right)^2 \\ &= \frac{1}{N^g} \sum_g \sum_{jk} \Delta J_{ij}^g \Delta J_{ik}^g \langle \sigma_j \sigma_k \rangle_g. \end{aligned}$$

It can be proved that the mean field technique is exact for long-range systems. This justifies a decorrelation between the dynamic variables σ_j and σ_k :

$$\langle \sigma_j \sigma_k \rangle_g = \langle \sigma_j \rangle_g \langle \sigma_k \rangle_g.$$

Then

$$\begin{aligned} \overline{\langle \Delta h_i^2 \rangle} &= \frac{1}{N^g} \sum_g \sum_{jk} \Delta J_{ij}^g \Delta J_{ik}^g \langle \sigma_j \rangle_g \langle \sigma_k \rangle_g \\ &= \frac{1}{N^g} \sum_g \sum_j \frac{J_1}{N} \langle \sigma_j \rangle_g^2 = J_1 Q, \end{aligned}$$

with

$$Q = \frac{1}{N} \sum_j \overline{\langle \sigma_j \rangle^2}.$$

Q is the Edwards Anderson (E.A.) parameter.

The average activity M is therefore given by

$$M = \frac{1}{\sqrt{2\pi}} \int dz \exp(-\frac{1}{2}z^2) S[\beta(z\sqrt{J_1 Q} + J_0 M)] \quad (4.31)$$

and the EA parameter Q by

$$Q = \frac{1}{\sqrt{2\pi}} \int dz \exp(-\frac{1}{2}z^2) S^2[\beta(z\sqrt{J_1 Q} + J_0 M)]. \quad (4.32)$$

The coupled equations 4.31 and 4.32 are the SK equations. There exists a range of temperature β^{-1} with $Q \neq 0$. Non-zero EA parameters Q are interpreted as being the sign that a new phase, the spin glass phase, appears in the system. In a spin glass phase the individual states σ_i fluctuate around *local* non-zero average values $\langle \sigma_i \rangle$ and these average values are distributed with a mean square deviation, which is Q .

4.3.3 The Hopfield model

The local field on a neuron i has been written as a sum of partial fields, one for every memorized pattern I^μ :

$$h_i = \sum_{\mu=1}^P h_i^\mu,$$

with

$$h_i^\mu = \xi_i^\mu M^\mu, \quad M^\mu = \frac{1}{N} \sum_j \xi_j^\mu \sigma_j.$$

In the process of ensemble averaging, all patterns except the first one I^1 are reshuffled, which makes up the constraint \mathcal{C}^g that the ensemble must obey. Then the first moment of the local field distribution is given by

$$\overline{\langle h_i \rangle} = \sum_{\mu=1}^P \overline{\langle h_i^\mu \rangle} = \overline{\langle h_i^1 \rangle} = \xi_i^1 M,$$

where we set $\langle M^1 \rangle = M$ for the sake of simplicity. A ‘local gauge transform’, $M \mapsto \xi_i^1 M$, makes M coincide with $\langle M^1 \rangle$. The second moment is

$$\begin{aligned} \overline{\langle \Delta h_i \rangle^2} &= \overline{\langle h_i \rangle^2} - (\overline{\langle h_i \rangle})^2 = \sum_{\mu>1} \overline{\langle h_i^\mu \rangle^2} \\ &= \sum_{\mu>1} \overline{(\xi_i^\mu M^\mu)^2} = \sum_{\mu>1} \overline{(M^\mu)^2} = R, \end{aligned}$$

which defines *the AGS (Amit, Guttmann and Sompolinsky) parameter R* .

The average activity is therefore given by

$$\overline{\langle \sigma_i \rangle} = \frac{1}{\sqrt{2\pi}} \int dz \exp(-\frac{1}{2}z^2) \mathcal{S}[\beta(z\sqrt{R} + \xi_i^1 M)]$$

and the order parameter M by

$$M = \frac{1}{\sqrt{2\pi}} \int dz \exp(-\frac{1}{2}z^2) \frac{1}{N} \sum_i \xi_i^1 \mathcal{S}[\beta(z\sqrt{R} + \xi_i^1 M)],$$

which is the first of the AGS equation. Henceforth we take $\mathcal{S}(x) = \tanh(x)$ and make profit of the odd parity of this function. One has

$$\xi_i^1 \mathcal{S}[\beta(z\sqrt{R} + M \xi_i^1)] = \mathcal{S}[\beta((\xi_i^1 z)\sqrt{R} + M)].$$

The change of variables, $\xi_i^1 z \mapsto z$, leads to

$$M = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} dz \exp(-\frac{1}{2}z^2) \tanh[\beta(z\sqrt{R} + M)]. \quad (4.33)$$

The second equation, giving the EA parameter Q , is derived likewise:

$$Q = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{+\infty} dz \exp(-\frac{1}{2}z^2) \tanh^2[\beta(z\sqrt{R} + M)]. \quad (4.34)$$

The order parameters Q and R are related: we compute the polarization created by the contribution $\langle h_i^\mu \rangle$ to the fields $\langle h_i \rangle$. To first order in $\langle h_i^\mu \rangle$ the local polarization due to the pattern I^μ is given by

$$\begin{aligned}\langle \sigma_i \rangle_\mu &= \tanh \left[\beta \sum_\mu \langle h_i^\mu \rangle \right] = \tanh \left[\beta \left(\langle h_i^\mu \rangle + \sum_{\mu' \neq \mu} \langle h_i^{\mu'} \rangle \right) \right] \\ &\simeq \langle \sigma_i \rangle_0 + \beta \langle h_i^\mu \rangle \left(1 - \tanh^2 \left[\beta \sum_{\mu' (\neq \mu)} \langle h_i^{\mu'} \rangle \right] \right),\end{aligned}$$

where $\langle \sigma_i \rangle_0$ is the activity of neuron i in the absence of the pattern I^μ . Then, using the relations $\langle h_i^\mu \rangle = \xi_i^\mu M^\mu$ and $M^\mu = (1/N) \sum_i \xi_i^\mu \langle \sigma_i \rangle$, the fluctuation M^μ is given by

$$M^\mu = \frac{1}{N} \sum_i \xi_i^\mu \langle \sigma_i \rangle_0 + \frac{\beta M^\mu}{N} \sum_i \left(1 - \tanh^2 \left[\beta \sum_{\mu' \neq \mu} \langle h_i^{\mu'} \rangle \right] \right).$$

This expression is squared and the sample average is carried out:

$$\overline{M^{\mu^2}} \left(1 - \frac{\beta}{N} \sum_i \left(1 - \tanh^2 [\beta \langle h_i \rangle] \right) \right)^2 = \frac{1}{N^2} \sum_i \overline{\langle \sigma_i \rangle^2} = \frac{1}{N} Q.$$

Since $N^{-1} \sum_i \overline{\left(1 - \tanh^2 [\beta \langle h_i \rangle] \right)} = 1 - N^{-1} \sum_i \overline{\langle \sigma_i \rangle^2} = 1 - Q$, one finds

$$R = \sum_\mu \overline{(M^\mu)^2} = \frac{\alpha Q}{(1 - \beta(1 - Q))^2}, \quad \text{where } \alpha = \frac{P}{N}. \quad (4.35)$$

The AGS equations (4.33), (4.34) and (4.35) which couple the three order parameters M , Q and R solve the Hopfield model for any number P of patterns and any level of noise β^{-1} . Clearly the order parameter M describes how well the pattern I^1 is retrieved. The physical meaning of order parameter Q is similar to the one it has in spin glasses: it accounts for the thermal fluctuations of states around their local fields. Finally, the order parameter R describes the fluctuations of states brought about by all memorized patterns except I^1 . The phase diagram, in the $(\alpha = P/N, \beta^{-1})$ space, which corresponds to these equations is determined numerically. It is displayed in Fig. 4.8. Results can be derived analytically along the β^{-1} axis, however. They are discussed in the next section.

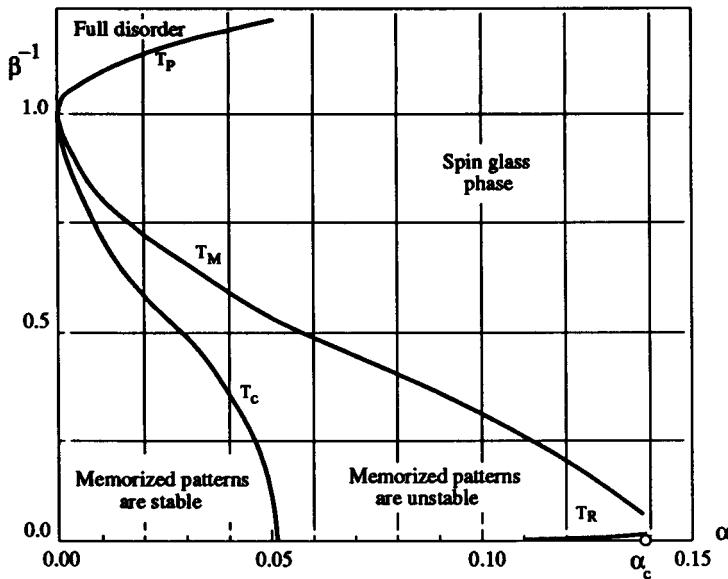


Figure 4.8. Phase diagram of a Hopfield neural network. The figure shows the three possible ordered states of the network: T_M is the boundary between a spin glass phase and a phase where memorization is achieved. Above T_c the stored patterns cease to be the ground states. They become less stable than spurious states. T_R is the line where the replica symmetry breaks down (After Amit, Gutfreund and Sompolinsky).

4.3.4 Memory storage capacities of the Hopfield model

The memory storage capacity of the Hopfield model can be derived from the study of the AGS equations in the limit of zero noise:

In the limit $\beta^{-1} = 0$, the tanh function becomes a sign function:

$$\tanh(x) \mapsto \text{sign}(x) = \begin{cases} +1 & \text{if } x > 0, \\ -1 & \text{if } x < 0. \end{cases}$$

The first equation becomes

$$\begin{aligned} M &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} dz \left[\exp\left(-\frac{1}{2}z^2\right) \left(\text{sign}\left(z + \frac{M}{\sqrt{R}}\right) + 1 - 1 \right) \right] \\ &= \frac{2}{\sqrt{2\pi}} \int_{-M/\sqrt{R}}^{+\infty} dz \exp\left(-\frac{1}{2}z^2\right) - 1 \\ &= \frac{2}{\sqrt{\pi}} \int_0^{M/\sqrt{2R}} dy \exp(-y^2) \end{aligned}$$

or

$$M = \operatorname{erf}\left(\frac{M}{\sqrt{2R}}\right). \quad (4.36)$$

On the other hand the expansion, to first order in β^{-1} , of the EA parameter is given by

$$\begin{aligned} Q &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} dz \left[\exp\left(-\frac{1}{2}z^2\right) \left(\tanh^2(\beta(z\sqrt{R} + M)) - 1 + 1 \right) \right] \\ &= 1 - \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} dz \frac{\exp\left(-\frac{1}{2}z^2\right)}{\cosh^2[\beta(z\sqrt{R} + M)]} \\ &= 1 - \frac{1}{\beta\sqrt{2\pi R}} \exp\left(-\frac{M^2}{2R}\right) \int_{-\infty}^{+\infty} dy \exp\left[-\frac{y^2 - 2\beta My}{2\beta^2 R}\right] \frac{1}{\cosh^2 y} \end{aligned}$$

and

$$Q = 1 - \frac{1}{\beta} \sqrt{\frac{2}{\pi R}} \exp\left(-\frac{M^2}{2R}\right) \quad (4.37)$$

since $\exp\left[-\frac{y^2 - 2\beta My}{2\beta^2 R}\right] = 1$ in the limit $\beta^{-1} = 0$. As $Q(\beta^{-1} = 0) = 1$ Eqs (4.35) and (4.37) yield

$$R \left(1 - \sqrt{\frac{2}{\pi R}} \exp\left[-\frac{M^2}{2R}\right] \right)^2 = \alpha, \quad (4.38)$$

that is,

$$\sqrt{R} = \sqrt{\alpha} + \sqrt{\frac{2}{\pi}} \exp\left[-\frac{M^2}{2R}\right].$$

One sets $Y = \frac{M}{\sqrt{2R}}$ or $Y\sqrt{2R} = \operatorname{erf}(Y)$ by using Eq. (3.91) to obtain the following equation:

$$\operatorname{erf}(Y) = Y \left(\frac{2}{\sqrt{\pi}} \exp(-Y^2) + \sqrt{2\alpha} \right).$$

This equation has always a solution $M = 0$, which is the spin glass solution. For low values of α , this solution is unstable. Another solution does exist with $M \neq 0$ (the patterns are retrieved), and is stable as long as

$$\frac{P}{N} = \alpha < \alpha_c = 0.138$$

For this value $M = 0.967$. This means that the retrieval brings about patterns comprising less than 2% wrong bits even for α s close to the

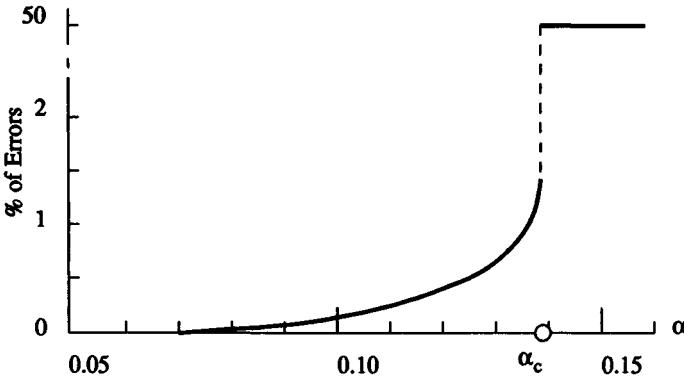


Figure 4.9. The figure shows the retrieval efficiency of a symmetrical noiseless Hebbian neural network. The maximum storage capacity is $\alpha_c = 0.138$. The diagram shows that retrieved patterns never display more than 2% errors. Retrieval properties steeply deteriorate however at α_c (After Amit, Gutfreund and Sompolinsky).

maximum capacity (see Figs 4.8 and 4.9). Stability cannot be discussed in the framework of the present approach. A study of stability requires other computational techniques such as the replica method, which is explained in detail in section 4.4.

Numerical simulations confirm the analytical results with two exceptions:

- a) The memorization efficiency deteriorates for values of α slightly greater than 0.138. This discrepancy is well explained by refinements of the replica theory.
- b) Memorization does not disappear completely above α_c . As Kinzel has pointed out, this remanence could be explained by the existence of correlations between the interactions which do not exist in general spin glasses and are inherent to the Hebbian models: in particular correlations along loops comprising an odd number of sites do not average to zero. For example:

$$N^3 \overline{J_{ij} J_{jk} J_{ki}} = \overline{\sum_{\mu} 1 + \sum_{\mu, \mu', \mu''} \xi_i^{\mu} \xi_j^{\mu} \xi_j^{\mu'} \xi_k^{\mu'} \xi_k^{\mu''} \xi_i^{\mu''}} = P.$$

The second sum is over all triples where μ , μ' and μ'' are different.

4.3.5 Storage capacities with asymmetrical Hebbian rules

Symmetry of synaptic connections is certainly a hypothesis which is not biologically founded. In this section we consider general asymmetrical Hebbian learning rules,

which, according to section 2.4.5, can be written as

$$J_{ij} = \frac{1}{N} \left[A \sum_{\mu=1}^P \xi_i^\mu \xi_j^\mu + B \sum_{\mu=1}^P \xi_i^\mu + C \sum_{\mu=1}^P \xi_j^\mu + PD \right].$$

Owing to the asymmetry of connections the dynamics of the network are not driven by an energy function. The techniques of statistical mechanics are therefore useless but the technique of local fields distribution is still applicable. The local fields are given by

$$h_i^\mu = A \xi_i^\mu M^\mu + B \xi_i^\mu M^0 + C M^\mu + N h_i^0, \quad \mu \neq 0,$$

with

$$M^0 = \frac{1}{N} \sum_j \langle \sigma_j \rangle, \quad h_i^0 = \alpha D M^0, \quad \alpha = \frac{P}{N}.$$

There are now two contributions to the field. One, h^1 , is associated with a particular pattern I^1 and the other, h^0 , with a uniform activity. Therefore there are four order parameters, Q , R , M^0 and M^1 . The last two order parameters are conjugate with h^0 and h^1 . The mean local field is given by

$$\langle h_i \rangle = M^1 (A \xi_i^1 + C) + M^0 (B \xi_i^1 + P D)$$

and the mean square deviation by

$$\langle h_i \rangle^2 = \lambda R + \nu (M^0)^2, \quad \lambda = A^2 + C^2,$$

$$\nu = B^2 P, \quad R = \sum_{\mu>1} \langle M^\mu \rangle^2.$$

The sample averaged activity of neuron i is then determined by

$$\begin{aligned} \langle \sigma_i \rangle &= \frac{1}{\sqrt{2\pi}} \int dz \exp\left(-\frac{1}{2}z^2\right) \\ &\times S \left[\beta \left(M^0 (B \xi_i^1 + P D) + M^1 (A \xi_i^1 + C) + z \sqrt{\lambda R + \nu (M^0)^2} \right) \right]. \end{aligned}$$

One deduces the equations for M^0 , M^1 and Q :

$$\begin{aligned} M^{(1)}_0 &= \frac{1}{\sqrt{2\pi}} \int dz \exp\left(-\frac{1}{2}z^2\right) \\ &\times \left(S \left[\beta \left(S M^0 + T M^1 + z \sqrt{\lambda R + \nu (M^0)^2} \right) \right] \right. \\ &\quad \left. + S \left[\beta \left(U M^0 + V M^1 + z \sqrt{\lambda R + \nu (M^0)^2} \right) \right] \right). \end{aligned}$$

On the other hand:

$$\begin{aligned} Q &= \frac{1}{2\sqrt{2\pi}} \int dz \exp\left(-\frac{1}{2}z^2\right) \\ &\times \left(S^2 \left[\beta \left(S M^0 + T M^1 + z \sqrt{\lambda R + \nu (M^0)^2} \right) \right] \right. \\ &\quad \left. + S^2 \left[\beta \left(U M^0 + V M^1 + z \sqrt{\lambda R + \nu (M^0)^2} \right) \right] \right), \end{aligned}$$

with

$$S = B + PD, \quad T = C + A,$$

$$U = -B + PD, \quad V = C - A,$$

and, finally, by using the lines of reasoning which we considered in the Hopfield model, we find a relation between M^0 , Q and R :

$$R = \frac{\alpha Q - P\beta^2 B^2 (M^0)^2 (1 - Q)^2}{(1 - \beta A(1 - Q))^2 + \beta^2 C^2 (1 - Q)^2}.$$

We will not go into the details of these complicated equations but we can give indications regarding the scaling of the memory storage capacity in the zero noise limit:

- The networks have no memorization capacities if the parameter A is negative. This can be understood by realizing that negative A s correspond to *antilearning processes*.
- The memory storage dwindles when $|C| > A$. If $|C| < A$ the memory capacities are preserved.
- A positive value of D limits the capacity to $P_c \simeq \sqrt{N}$. The occurrence of positive values of D is therefore very damaging. Negative D s however are innocuous. One could imagine that the effect of (inhibitory) interneurons would be to yield an overall negative value of D , so saving the memory capacities of the network.
- The parameter B does not change the scaling of the usual symmetrical Hopfield models $P_c \simeq N$.

To summarize: *the main properties of symmetrically connected Hebbian nets are not spoiled by large fluctuations of learning rule parameters.*

4.4 The replica method approach

Let us now turn to the derivation of the AGS equations using the replica technique. One of the merits of this approach is to provide criteria, based upon the study of free energy, for the stability of the phases which the network can display. It is also possible with this technique to go beyond the approximations we have used up to now by breaking the symmetry of replicas in a way similar to that used in the theory of spin glasses. These complications do not greatly change the conclusions one arrives at using the replica symmetry hypothesis, and they are only briefly mentioned at the end of this section.

In the framework of statistical mechanics ($S(x) = \tanh(x)$), the quantity to be computed is the sample average \bar{F} of free energy because F is an extensive quantity of the system, which means that it is proportional to the size N of the network. The partition function, Z , for example, is not an extensive quantity. According to statistical mechanics one has

$$\bar{F} = -\frac{1}{\beta} \log \bar{Z} = -\frac{1}{\beta} \sum_{\{\xi_i^\mu\}} P(\{\xi_i^\mu\}) \log(Z(\{\xi_i^\mu\})).$$

It is difficult to compute the averaged logarithm. One then uses the relations

$$Z^n = \exp(n \log Z) \simeq 1 + n \log Z$$

in the limit of small ns , whence

$$\log(Z) = \lim_{n \rightarrow 0} \frac{Z^n - 1}{n}. \quad (4.39)$$

The computation of $\overline{Z^n}$ is much easier than that of $\overline{\log Z}$. Z^n is given by

$$Z^n = \prod_{a=1}^n Z(\{\sigma_i^a\}) = \sum_{\{\sigma_i^a\}} \prod_{a=1}^n \prod_{\mu=1}^P \exp \left[+\frac{\beta}{2N} \left(\sum_i \xi_i^\mu \sigma_i^a \right)^2 \right],$$

where $a = 1, \dots, n$ labels a set of n systems, or *replicas*, identical to the original system.

The derivation of the free energy is decomposed into several steps.

a) *Elimination of disorder*

One first uses a Gaussian transform to linearize the exponent with respect to the dynamical variables σ_i^a (see section 4.2.2):

$$Z^n = \sum_{\{\sigma_i^a\}} \prod_{a=1}^n \prod_{\mu=1}^P \left(\frac{N\beta}{2\pi} \right)^{1/2} \int_{-\infty}^{+\infty} dM_a^\mu$$

$$\times \exp \left[-\frac{1}{2} N \beta (M_a^\mu)^2 + \beta M_a^\mu \left(\sum_i \xi_i^\mu \sigma_i^a \right) \right].$$

The sample average is carried out as usual, that is by renewing all memorized patterns I^μ *except the first one* I^1 , which is left unchanged. The formalism can also tackle more general cases, but the introduction of the then necessary supplementary indices would damage the transparency of the proof. The contribution of M^1 to Z^n is therefore singled out:

$$Z^n = \sum_{\{\sigma_i^a\}} \left(\frac{N\beta}{2\pi} \right)^{nP/2}$$

$$\times \int d\tilde{M}_a \exp \left[N\beta \left(-\frac{1}{2} \sum_{\mu \neq 1} \sum_a (M_a^\mu)^2 + \frac{1}{N} \sum_{\mu \neq 1} \sum_a M_a^\mu \left(\sum_i \xi_i^\mu \sigma_i^a \right) \right) \right]$$

$$\times \int dM_a^1 \exp \left[N\beta \left(-\frac{1}{2} \sum_a (M_a^1)^2 + \frac{1}{N} \sum_a M_a^1 \left(\sum_i \xi_i^1 \sigma_i^a \right) \right) \right].$$

It is now easy to carry out the average on disorder since the exponents of the partition function are linear functions of the random variables ξ_i^μ , $\mu = 2, 3, \dots, P$. This possibility is the main interest of the replica technique. One obtains

$$\int d\xi_i^\mu P(\xi_i^\mu) \exp \left(\beta \sum_{a=1}^n M_a^\mu \xi_i^\mu \sigma_i^a \right)$$

$$= \frac{1}{2} \left(\exp \left[\beta \sum_a M_a^\mu \sigma_i^a \right] + \exp \left[-\beta \sum_a M_a^\mu \sigma_i^a \right] \right)$$

$$= \exp \left[\log \left(\cosh \left(\beta \sum_a M_a^\mu \sigma_i^a \right) \right) \right].$$

However, all the order parameters M^μ are of the order of $1/\sqrt{N}$, except M^1 , which is of the order of 1. Therefore it is legitimate to expand the exponent according to

$$\log \cosh(x) \simeq \log\left(1 + \frac{1}{2}x^2\right) \simeq \frac{1}{2}x^2,$$

which yields

$$\prod_{\mu=1}^P \int d\xi_i^\mu \dots = \exp\left[\frac{1}{2}\beta^2 \sum_\mu \sum_a \sum_b M_a^\mu M_b^\mu \sigma_i^a \sigma_i^b\right],$$

and finally the disorder averaged partition function is given by

$$\begin{aligned} \overline{Z^n} &= \sum_{\{\sigma_i^a\}} \left(\frac{N\beta}{2\pi}\right)^{Pn/2} \\ &\times \int \mathcal{D}\tilde{M}_a \exp\left[N\beta\left(-\frac{1}{2} \sum_{\mu \neq 1} \sum_a \left(M_a^\mu\right)^2 + \frac{\beta}{2N} \sum_{\mu \neq 1} \sum_{i,a,b} M_a^\mu M_b^\mu \sigma_i^a \sigma_i^b\right)\right] \\ &\times \int dM_a^1 \exp\left[N\beta\left(-\frac{1}{2} \sum_a (M_a^1)^2 + \frac{1}{N} \sum_a \sum_i M_a^1 \xi_i^1 \sigma_i^a\right)\right]. \end{aligned}$$

b) *Introduction of the AGS and EA order parameters R_{ab} and Q_{ab} and elimination of small-order parameters*

The order parameters M_a^μ , ($\mu > 1$) are of the order of $1/\sqrt{N}$, but the first integral contains the terms

$$\sum_{\mu=2}^P M_a^\mu M_b^\mu,$$

which are of the order of 1 when P is of the order of N , that is of the order of M^1 . These terms are therefore also order parameters R_{ab} which must be fixed in the process of integration. This is achieved by introducing Lagrange multipliers Q_{ab} . Since $R_{ab} \equiv R_{ba}$ it is necessary to sum up the parameters R_{ab} with $a < b$. Using the results of section 4.2.2, the first integral of $\overline{Z^n}$ thus becomes

$$\begin{aligned} &\int \mathcal{D}\tilde{M}_a \prod_{a < b} \int dR_{ab} \prod_{a < b} \int dQ_{ab} \\ &\quad \times \exp\left[N\beta\left(-\frac{1}{2}(1-\beta) \sum_{\mu \neq 1, a} (M_a^\mu)^2 + \frac{\beta}{2N} \sum_{i,a,b} R_{ab} \sigma_i^a \sigma_i^b\right)\right] \\ &\quad \times \exp\left[\frac{1}{2}N\beta^2 \left(\sum_{a,b} Q_{ab} \left(\sum_{\mu \neq 1} M_a^\mu M_b^\mu - R_{ab}\right)\right)\right]. \end{aligned}$$

This expression is rewritten as

$$\begin{aligned} &\prod_{a < b} \int dR_{ab} \int dQ_{ab} \\ &\times \exp\left[-\frac{1}{2}N\beta^2 \sum_{a,b} Q_{ab} R_{ab} + \frac{1}{2}\beta^2 \sum_{i,a,b} R_{ab} \sigma_i^a \sigma_i^b\right] \\ &\times \prod_{\mu,a} \int dM_a^\mu \exp\left[N\beta\left(-\frac{1}{2}(1-\beta) \sum_a (M_a^\mu)^2 + \frac{1}{2}\beta \sum_{a,b} Q_{ab} M_a^\mu M_b^\mu\right)\right] \end{aligned}$$

and the small observables M_a^μ ; ($\mu > 1$) are summed out. The corresponding integration is classical. Using

$$\int_{-\infty}^{+\infty} dx \exp\left[-\frac{1}{2}N\beta\lambda_a x^2\right] = \left(\frac{2\pi}{N\beta\lambda_a}\right)^{1/2},$$

one finds that the result is proportional to the product of inverses of the square roots of the eigenvalues λ_a of the matrix:

$$\mathbf{X} = (1 - \beta) \mathbf{I} - \beta \mathbf{Q},$$

where \mathbf{I} is the unit matrix and \mathbf{Q} is an $n \times n$ matrix with elements $(\mathbf{Q})_{ab} = Q_{ab}$, $(\mathbf{Q})_{aa} = 0$. This product can be written as

$$\begin{aligned} \prod_{a=1}^n \lambda_a^{-1/2} &= \prod_a \exp\left(-\frac{1}{2} \log \lambda_a\right) \\ &= \exp\left[-\frac{1}{2} \sum_a \log(\lambda_a)\right] = \exp\left(-\frac{1}{2} \text{Tr} \log \mathbf{X}\right) \end{aligned}$$

and the disorder averaged partition function becomes

$$\begin{aligned} \overline{Z^n} &= \sum_{\{\sigma_i^a\}} \prod_{a < b} \int dR_{ab} \prod_{a < b} \int dQ_{ab} \prod_a \int dM_a^1 \\ &\quad \times \exp\left[N\beta\left(-\frac{1}{2} \sum_a (M_a^1)^2 - \frac{1}{2}\beta \sum_{a,b} Q_{ab} R_{ab} \right.\right. \\ &\quad \left.\left. - \frac{P}{2N\beta} \text{Tr} \log((1 - \beta)\mathbf{I} - \beta\mathbf{Q}) + \frac{1}{N} \sum_a M_a^1 \sum_i \xi_i^1 \sigma_i^a + \frac{\beta}{2N} \sum_{i,a,b} R_{ab} \sigma_i^a \sigma_i^b\right)\right]. \end{aligned}$$

c) *Computation of the free energy, the replica symmetry and the steepest gradient approximation*

In the *replica symmetry approximation* all order parameters are reduced to three order parameters:

$$M = M_a^1, \quad \text{the 'magnetization' parameter};$$

$$R = R_{ab}, \quad \text{the AGS parameter};$$

$$Q = Q_{ab}, \quad \text{the EA parameter}.$$

Then using the steepest descent (or saddle point) approximation one obtains an expression of the free energy:

$$\begin{aligned} \frac{\overline{F^n}}{N} &= \frac{1}{2} \sum_a M^2 + \frac{1}{2}\beta \sum_a \sum_{b \neq a} QR \\ &\quad + \frac{\alpha}{2\beta} \text{Tr} \log[(1 - \beta)\mathbf{I} - \beta\mathbf{Q}] - \frac{1}{N\beta} \log \sum_{\{\sigma_i^a\}} \exp[\beta H(\{\sigma_i^a\})], \end{aligned}$$

where $\alpha = P/N$, and

$$H(\{\sigma_i^a\}) = \sum_i H_i(\{\sigma_i^a\}) = \sum_i \left(\frac{1}{2} \beta \sum_a \sum_{b \neq a} R \sigma_i^a \sigma_i^b + \sum_a M \sigma_i^a \xi_i^1 \right).$$

Let us compute the various terms of the free energy in the replica symmetry hypothesis. One has

$$\frac{1}{2} \sum_a M^2 = \frac{1}{2} n M^2, \quad \frac{1}{2} \beta \sum_a \sum_{b \neq a} Q R = \frac{1}{2} \beta Q R (n^2 - n).$$

The matrix \mathbf{X} has

$$\begin{cases} \text{one eigenvalue: } 1 - \beta - (n-1)\beta Q, \\ (n-1) \text{ eigenvalues: } 1 - \beta + \beta Q, \end{cases}$$

and therefore

$$\begin{aligned} \text{Tr} \log[(1-\beta)\mathbf{I} - \beta \mathbf{Q}] &= (n-1) \log(1 - \beta + \beta Q) \\ &\quad + \log(1 - \beta - (n-1)\beta Q) \\ &\simeq n \left[\log(1 - \beta + \beta Q) - \frac{\beta Q}{1 - \beta + \beta Q} \right]. \end{aligned}$$

Finally one computes the last term. One has

$$\begin{aligned} I_i &= \sum_{\{\sigma_i^a\}} \exp[\beta H_i(\{\sigma_i^a\})] \\ &= \sum_{\{\sigma_i^a\}} \exp \left[\frac{1}{2} \beta^2 R \left(\sum_a \sum_b \sigma_i^a \sigma_i^b - n \right) + \beta M \sum_a \sigma_i^a \xi_i^1 \right] \\ &= \exp(-\frac{1}{2} \beta^2 R n) \sum_{\{\sigma_i^a\}} \exp \left[\frac{1}{2} \beta^2 R \left(\sum_a \sigma_i^a \right)^2 + \beta M \sum_a \sigma_i^a \xi_i^1 \right] \end{aligned}$$

and one uses the Gaussian transform once more:

$$\exp \left[\frac{1}{2} R \beta^2 \left(\sum_a \sigma_i^a \right)^2 \right] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} dz \exp \left[-\frac{1}{2} z^2 + \beta z \sqrt{R} \left(\sum_a \sigma_i^a \right) \right].$$

Then

$$\begin{aligned} I_i &= \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2} \beta^2 R n) \sum_{\{\sigma_i^a\}} \prod_a \int dz \exp \left[-\frac{1}{2} z^2 + \beta z \sqrt{R} \sigma_i^a + \beta M \sigma_i^a \xi_i^1 \right] \\ &= \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2} \beta^2 R n) \int dz \exp(-\frac{1}{2} z^2) \left[2 \cosh \left(\beta (z \sqrt{R} + M \xi_i^1) \right) \right]^n, \end{aligned}$$

which yields

$$\begin{aligned} \frac{\overline{F^n}}{N} &= \frac{1}{2}nM^2 + \frac{1}{2}\beta Q R(n^2 - n) \\ &\quad + \frac{\alpha n}{2\beta} \left[\log(1 - \beta + \beta Q) - \frac{\beta Q}{1 - \beta + \beta Q} \right] \\ &\quad + \frac{1}{2}\beta Rn - \frac{n}{\beta N} \sum_i \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} dz \exp\left(-\frac{1}{2}z^2\right) \\ &\quad \times \left[\log 2\cosh\left(\beta z\sqrt{R} + M\xi_i^1\right) \right], \end{aligned}$$

since, in the limit $n \rightarrow 0$, one has

$$\log \int dz \exp\left(-\frac{1}{2}z^2\right) f^n(z) \simeq n \int dz \exp\left(-\frac{1}{2}z^2\right) \log[f(z)].$$

Taking the limit Eq. (4.39),

$$\lim_{n \rightarrow 0} \frac{\overline{F^n}}{nN},$$

one finally obtains (the bar indicates an average over sites)

$$\begin{aligned} \bar{f} &= \frac{1}{2}M^2 + \frac{\alpha}{2\beta} \left[\log(1 - \beta + \beta Q) - \frac{\beta Q}{1 - \beta + \beta Q} \right] + \frac{1}{2}\beta R(1 - Q) \\ &\quad - \frac{1}{\beta\sqrt{2\pi}} \int dz \exp\left(-\frac{1}{2}z^2\right) \left[\log 2\cosh\left(\beta(z\sqrt{R} + M\xi^1)\right) \right]. \end{aligned}$$

d) Order parameters

These are determined by minimizing the free energy. From $\partial\bar{f}/\partial M = 0$ one obtains

$$\begin{aligned} M &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} dz \exp\left(-\frac{1}{2}z^2\right) \frac{1}{N} \sum_i \xi_i \tanh\left(\beta(z\sqrt{R} + M\xi_i^1)\right) \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} dz \exp\left(-\frac{1}{2}z^2\right) \frac{1}{N} \sum_i \tanh\left(\beta(z\sqrt{R}\xi_i^1 + M)\right), \end{aligned}$$

which, with the change of variable $z\xi_i^1 \mapsto z$, becomes

$$M = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} dz \exp\left(-\frac{1}{2}z^2\right) \tanh\left(\beta(z\sqrt{R} + M)\right). \quad (4.40)$$

Similarly, from $\partial\bar{f}/\partial R = 0$ and a few analytical calculations:

$$Q = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} dz \exp\left(-\frac{1}{2}z^2\right) \tanh^2\left(\beta(z\sqrt{R} + M)\right) \quad (4.41)$$

and

$$R = \frac{\alpha Q}{(1 - \beta + \beta Q)^2} \quad (4.42)$$

from $\partial \tilde{f} / \partial Q = 0$. The meanings of the order parameters are inferred from the saddle point equations:

$$\begin{aligned} M &= \frac{1}{N} \sum_i \xi_i^1 \langle \sigma_i \rangle = \overline{\langle \sigma_i \rangle}, \\ Q &= \frac{1}{N} \sum_i \langle \sigma_i \rangle^2 = \overline{\langle \sigma_i \rangle^2}, \\ R &= \sum_{\mu \neq 1} \left(\frac{1}{N} \sum_i \xi_i^\mu \langle \sigma_i \rangle \right)^2 = \sum_{\mu \neq 1} \overline{(M^\mu)^2}. \end{aligned}$$

e) *Stability of states*

At low noise levels, ($\beta^{-1} = 0$), the energy is given by

$$\begin{aligned} \tilde{f} &= -\frac{1}{2N^2} \sum_\mu \sum_i \sum_j \xi_i^\mu \xi_j^\mu \sigma_i \sigma_j \\ &= -\frac{1}{2N^2} \left[\left(\sum_i \xi_i^1 \sigma_i \right)^2 + \sum_{\mu \neq 1} \left(\sum_i \xi_i^\mu \sigma_i \right)^2 \right] \\ &= -\frac{1}{2} (M^2 + R). \end{aligned}$$

In the spin glass phase $M = 0$ and therefore, according to Eq. (4.38),

$$R \left(1 - \sqrt{\frac{2}{\pi R}} \right)^2 = \alpha \implies R = (\sqrt{\alpha} + \sqrt{2/\pi})^2.$$

Comparing the energy \tilde{f}^{SG} of the spin glass phase ($M = 0$) with that of the retrieval phase \tilde{f}^M ($M = 0.967$) in the vicinity of the critical capacity $\alpha_c = 0.138$, one finds

$$\tilde{f}^{SG} - \tilde{f}^M = -0.117.$$

Therefore the spin glass phase is the most stable one. The relative stabilities of the two phases is inverted when $\alpha < 0.051$. This inversion is made conspicuous by building the histogram of the energies of the metastable states of the system (see Fig. 4.3). For values of α ranging from 0.051 to 0.138 the memorized states are irreversibly destabilized by large enough noises.

The transition line between the paramagnetic (fully disordered) and the spin glass phases can be deduced from the following relations, which are derived from Eqs (4.41) and (4.42) in the limit $M = 0$, $Q, R \rightarrow 0$, $\beta \rightarrow 1$:

$$Q \simeq \beta^2 R \simeq \frac{\alpha Q}{(1 - \beta)}.$$

It is given by (see Fig. 4.8)

$$\beta_{SG}^{-1} \simeq 1 + \sqrt{\alpha}. \quad (4.43)$$

Remark

This analysis has been pushed a step further by Crisanti *et al.*, who appealed to the scheme of *replica symmetry breaking* which has been introduced by Parisi in the

context of spin glasses. The idea is to introduce a hierarchical set of parameters $Q_x = Q, Q', Q'', \dots$, which gives more flexibility to the space of order parameters. Successive approximations consist in replacing the simple matrix

$$\mathbf{Q} = \begin{pmatrix} 0 & & & \\ \ddots & & & \\ & 0 & Q & \\ & Q & & \ddots \\ & & & 0 \end{pmatrix}$$

by

$$\mathbf{Q}' = \left(\begin{array}{cc|cc} 0 & Q & Q' & Q' \\ Q & 0 & Q' & Q' \\ \hline Q' & Q' & 0 & Q \\ Q' & Q' & Q & 0 \end{array} \right),$$

then by

$$\mathbf{Q}'' = \left(\begin{array}{cc|cc|cc} 0 & Q & Q' & Q' & & & \\ Q & 0 & Q' & Q' & & & \\ \hline Q' & Q' & 0 & Q & Q'' & & \\ Q' & Q' & Q & 0 & & & \\ \hline & & & & 0 & Q & \\ & & & & Q & 0 & \\ & & & & Q' & Q' & \\ & & & & Q' & Q' & \\ & & & & Q' & Q' & \\ & & & & Q' & Q' & \end{array} \right),$$

and so on. The results one obtains by using this technique are very close to those which are derived by keeping the symmetry of replicas. For example, the storage capacity is $\alpha_c = 0.145$ at the first stage of replica symmetry breaking, a slight increase with respect to the former value of $\alpha_c = 0.138$. The difference is nevertheless observed in numerical simulations.

As one can see, the theory of symmetrical Hebbian neural networks is rather well understood. A number of analytical results have been obtained and checked numerically. Even though the model is far from being realistic, it may be considered as a solid basis for developments to come. There remains a problem of paramount importance, however: that of confronting, at least qualitatively, the predictions with experimental observations. Unfortunately, experiments are difficult and sparse. A little information relating to the problem of the relevance of the theory is gathered in the last chapter, where it is used in a critical overview on the whole approach of the theory of neural networks which is proposed in this text.

4.5 General dynamics of neural networks

4.5.1 The problem

The theory of the dynamics of neural networks imprinted with any numbers of patterns is much harder than the study of its steady properties. In actual fact, at the time of the writing of this text, the problem had not been solved yet. The reason is the following:

Let us consider a neuron i of a network Σ^g . At time ν its activity $\langle \sigma_i(\nu) \rangle$ is given by a thermal average over an ensemble of equivalent networks, starting from the initial conditions $\{\sigma_i(\nu = 0)\}$. We have seen that $\sigma_i(\nu)$ is determined by the set of partial fields

$$h_i(\nu) = \sum_j J_{ij} \sigma_j(\nu - 1)$$

generated by the states $\sigma_j(\nu - 1)$ of the various neurons j neuron i is linked to. Let us assume that the states of neurons i and j , $\sigma_i(\nu - 1)$ and $\sigma_j(\nu - 1)$, at time $(\nu - 1)$ are correlated:

$$\langle \sigma_i(\nu - 1) \sigma_j(\nu - 1) \rangle \neq 0. \quad (4.44)$$

Then the fields $h_i(\nu)$ are correlated and the states $\sigma_i(\nu)$ and $\sigma_j(\nu)$ at time ν are also correlated. This means that correlations propagate along the whole history of the networks of the ensemble. As long as the number of patterns remains finite these correlations have negligible effects. However, when the number is of the order of N the correlations build up, which results in the introduction of the AGS order parameter R , as we saw. More specifically, the computation of $\langle \sigma_i(\nu) \rangle$ involves thermal averages of two-site correlations taken at time $(\nu - 1)$ as in Eq. (4.44). For the computation of these quantities to be carried out, averages of three-site correlations taken at time $(\nu - 2)$ are needed in turn and so on. These correlations make difficult the building of a general theory of the dynamics of neural networks. There exist neuronal architectures, however, in which correlations have no effect and where the general equations of the neuronal dynamics have been derived. The first are Hebbian diluted networks which have been studied by Derrida *et al.* and the others are Hebbian layered networks which have been discussed by Domany *et al.* We give only a brief account of these theories.

4.5.2 Hebbian diluted networks

In the model proposed by Derrida, Gardner and Zippelius a number of interactions J_{ij} are broken at random. As J_{ij} and J_{ji} can be broken separately, the result is an asymmetrically connected network. The remaining average connectivity is K (on the average a neuron is connected

to K neurons). The value of intact interactions is given by the usual Hebbian rule. Breaking the connections also breaks the chains of correlation functions one needs to compute the neuronal activities. For example, the calculation of $\langle \sigma_i(\nu) \rangle$ involves a tree of ancestors whose states at $\nu = 0$ influence σ_i at time ν . The span of the tree, that is to say the number of such ancestors, is of the order of K^ν . The number of ancestors that are common to two different trees is

$$K^\nu \left(\frac{K^\nu}{N} \right),$$

and therefore if

$$K^\nu < N^{1/2} \quad (4.45)$$

two neurons cannot have the same ancestor. The consequence is that the propagation of correlations along the history of the ensemble of networks can be ignored. The condition (4.45) is satisfied for all ν s if K scales as N^0 . Therefore dilution implies that

$$K < \log N. \quad (4.46)$$

One computes the second moment of the fields distribution created by the initial states on site i . One finds

$$\sum_{\mu>1}^P \langle h_i^\mu \rangle^2 = \frac{2KP}{N},$$

where the factor of 2 comes from the fact that one now distinguishes between the interactions J_{ij} and J_{ji} . Owing to the properties of diluted networks the second moment is left unchanged during the dynamics: the time evolution of local fields manifests itself by a mere shift of their Gaussian distribution. Then the equations of the (parallel) dynamics simply couple the overlap M ($\equiv M^1$) taken at two successive times:

$$M(\nu + 1) = \frac{1}{\sqrt{2\pi}} \int dz \exp(-\frac{1}{2}z^2) \mathcal{S}\left[\beta_K(z\sqrt{2\alpha} + M(\nu))\right], \quad (4.47)$$

$$\text{with } \alpha = \frac{P}{K} \left(= \frac{P}{\log N} \right) \text{ and } \beta_K = \beta \frac{K}{N}.$$

There exists a critical noise $\beta_K = 1$ where memory disappears ($M = 0$). But this is now a *smooth transition* (a second-order transition). At low noise level ($\beta_K^{-1} = 0$) the critical value is given by

$$\alpha_c = \frac{2}{\pi} = 0.637.$$

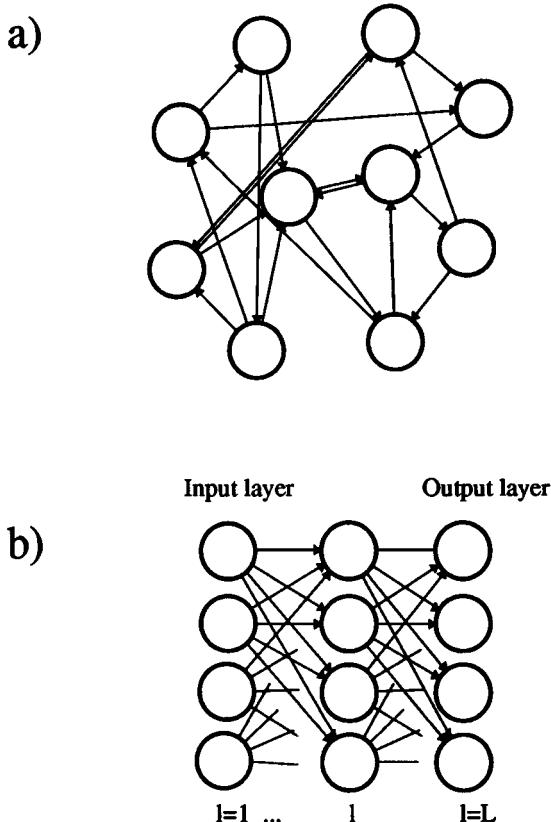


Figure 4.10. Neural network architectures whose dynamics can be studied analytically even in the limit of large numbers of patterns. a) Sparsely connected networks. b) Feedforward layered networks.

4.5.3 Hebbian layered networks

The architecture of layered networks is made of a series of sets of neurons or layers $\ell = 1, 2, \dots, L$, where all neurons of layer ℓ are connected to and only to all neurons of layer $\ell + 1$ (see Fig. 4.10).

In this model the layers are identical and the connections from layer ℓ to layer $\ell + 1$ are given by

$$J_{i \in \ell, j \in \ell+1} = \frac{1}{N_L} \sum_{\mu} \xi_{i \in \ell}^{\mu} \xi_{j \in \ell+1}^{\mu}.$$

N_L is the number of neurons in a layer.

There are no loops in these systems and the states of the neurons

belonging to one layer are fully determined by the states of the neurons belonging to the preceding layer. The polarization of the neurons proceeds layer by layer. The field distribution on a site of layer $\ell + 1$, its two first moments in particular, only depends on the field distributions on layer ℓ , exactly as the field distribution on site i at time $\nu + 1$ depends on the field distributions on other neurons at time ν (in parallel dynamics), but with the interesting feature that the neurons involved at this ‘time $\ell + 1$ ’ are different from those involved at ‘time ℓ ’. This simple remark shows that one can derive equations giving the average properties of the network by appealing to the analysis we used for the steady case and then transform the result into dynamical equations by simply using the correspondence $\ell \mapsto \nu$. The calculations of the field distribution follow the lines we exposed for the steady case, and the three AGS equations become

$$\begin{aligned} M(\ell + 1) &= \frac{1}{\sqrt{2\pi}} \int dz \exp(-\frac{1}{2}z^2) S[\beta(z\sqrt{R(\ell)} + M(\ell))], \\ Q(\ell + 1) &= \frac{1}{\sqrt{\pi}} \int dz \exp(-\frac{1}{2}z^2) S^2[\beta(z\sqrt{R(\ell)} + M(\ell))], \\ R(\ell + 1) &= \alpha Q(\ell) + R(\ell)(\beta(1 - Q(\ell)))^2. \end{aligned}$$

These equations have been derived by Meir and Domany starting directly from the Little dynamics. Their study shows that the dynamics may have a non-zero fixed point $M^* \neq 0$, depending on the values of the two parameters α and β of the model. Whether the fixed point is reached or not also depends on the initial overlap M ($\ell = 1$). One of the main points of interest with regard to dynamical equations is that they allow an analytical study of basins of attractions which completely escapes a theory of stationary systems: the size of the basins is determined by the critical initial overlaps which separate the trivial attractor $M^* = 0$ from the non-trivial attractor $M^* \neq 0$ when it exists. Many more results can be found in the papers by the authors quoted.

TEMPORAL SEQUENCES OF PATTERNS

Something essential is missing in the description of memory we have introduced in previous chapters. A neural network, even isolated, is a continuously evolving system which never settles indefinitely in a steady state. We are able to retrieve not only single patterns but also ordered strings of patterns. For example, a few notes are enough for an entire song to be recalled, or, after training, one is able to go through the complete set of movements which are necessary for serving in tennis. Several schemes have been proposed to account for the production of memorized strings of patterns. Simulations show that they perform well, but this does not mean anything as regards the biological relevance of the mechanisms they involve. In actual fact no observation supporting one or the other of the schemes has been reported so far.

5.1 Parallel dynamics

Up to now the dynamics has been built so as to make the memorized patterns the fixed points of the dynamics. Once the network settles in one pattern it stays there indefinitely, at least for low noise levels. We have seen that fixed points are the asymptotic behaviors of rather special neural networks, namely those which are symmetrically connected. In asymmetrically connected neural networks whose dynamics is deterministic and parallel (the Little dynamics at zero noise level), the existence of limit cycles is the rule. It is then tempting to imagine that the retrieval of temporal sequences of patterns occurs through limit cycles.

The memorization process then consists in making the patterns I^1, I^2, \dots, I^P of a sequence C identical to one of the limit cycles of the dynamics.

We assume that $\sigma_i(\nu) = \xi_i^\mu$. The state $\sigma_i(\nu + 1) = \xi_i^{\mu+1}$ is stable at time $\nu + 1$ if

$$\sigma_i(\nu + 1) h_i^\mu = \xi_i^{\mu+1} h_i^\mu > 0,$$

with

$$h_i^\mu(\nu) = \sum_j J_{ij} \sigma_j(\nu) = \sum_j J_{ij} \xi_j^\mu.$$

This can be achieved by modifying the Hebb rule so as to couple the patterns of the sequence

$$J_{ij} = \frac{1}{N} \sum_{\mu'=1}^P \xi_i^{\mu'+1} \xi_j^{\mu'}, \quad (5.1)$$

where the pattern I^{P+1} is identical to pattern I^1 . Indeed,

$$h_i^\mu = \sum_j \frac{1}{N} \sum_{\mu'} \xi_i^{\mu'+1} \xi_j^{\mu'} \xi_j^\mu = \xi_i^{\mu+1} + O(P/N^{1/2})$$

and $\xi_i^{\mu+1} h_i^\mu > 0$ if the system is not overcrowded (see Fig. 5.1).

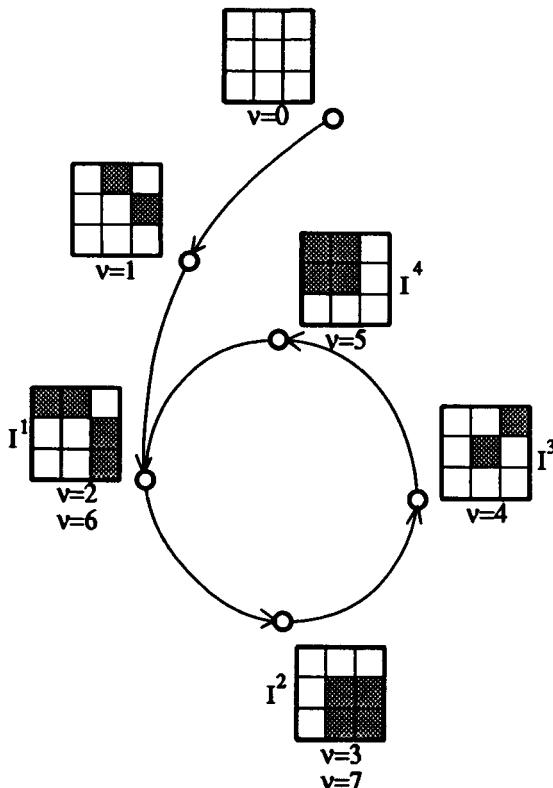


Figure 5.1. Retrieving a temporal sequence of patterns by making it a limit cycle of the parallel dynamics.

The learning rule (5.1) is valid for one cycle. It can be extended to many non-intersecting cycles C :

$$J_{ij} = \frac{1}{N} \sum_C \sum_{\mu' (\in C)} \xi_i^{\mu'+1,C} \xi_j^{\mu',C}.$$

We have seen that limit cycles exist only in the zero noise limit and for parallel dynamics. These two restrictions make it unlikely that biological systems produce temporal sequences that way. Simulations show that the dynamics are very sensitive to perturbations such as a slight lack of synchronism or light noises. In particular, the basins of attraction of large enough cycles are very narrow: flipping one or two states makes the trajectory escape the cycle. Finally, according to this mechanism, the flipping time is $\tau_r \simeq 4$ ms, which is much too short to account for the observations.

It is worth noting that rules similar to Eq. (5.1) can be applied to any strings of patterns C , not necessarily to loops, provided that the so defined paths (in the phase space) do not self-intersect.

5.2 Stochastic dynamics

5.2.1 Temporal sequences and synaptic efficacies

Other mechanisms rest upon asynchronous dynamics and are robust with respect to perturbations. All the following schemes rely upon two sorts of interactions:

a) A first one which tends to stabilize the patterns. It is of the usual symmetrical Hopfield type:

$$J_{ij}^s = \frac{1}{N} \sum_{\mu} \xi_i^{\mu} \xi_j^{\mu}.$$

b) The second, which acts as a pointer, tends to make the system jump from one state to the successor state in the sequence. It is analogous to the type we saw in the parallel dynamics:

$$J_{ij}^a = \frac{1}{N} \sum_{\mu} \xi_i^{\mu+1} \xi_j^{\mu}.$$

The overall synaptic efficacies are not symmetrical and, strictly speaking, it is no more possible to view the phase space of the system as an energy landscape. However, it helps to imagine that the state of the system wanders in a landscape determined by the first type of interactions and skewed by the second type.

The various mechanisms differ by the nature of the force which drives the system from one state to the other. The driving mechanism can be an external uniform field (Lehmann), noise (Buhmann *et al.*), delays (Sompolinsky *et al.*), synaptic plasticity (Peretto *et al.*) or threshold modifications (Horn *et al.*).

5.2.2 Sequences driven by an external field

It is obvious that a field given by

$$h_i = \xi_i^\mu$$

will drive the neural system in state I^μ whatever the current state. This is of no interest. More interesting is the jump of the system promoted by a uniform field, a signal which has no relationship with any of the patterns whatsoever. This is the case of counting chimes, for example (Amit): the system jumps from one pattern representing a number to another pattern representing the following number at every strike of a bell. In the mechanism proposed by Lehmann, the jump is driven by a simple uniform field h . This mechanism is equivalent to a uniform rise of the value of thresholds, that is to say, it is equivalent to the habituation phenomenon. This scheme is now described:

Let I^1, I^2, I^3 and I^4 be four orthogonal patterns. One also defines $I^{1.5}$ and $I^{1.75}$ as follows:

$$\xi_i^{1.5} = \begin{cases} \xi_i^2 & \text{if } \xi_i^1 = \xi_i^2, \\ 1 & \text{otherwise;} \end{cases}$$

$$\xi_i^{1.75} = \begin{cases} \xi_i^2 & \text{if } \xi_i^1 = \xi_i^2 (= \pm \xi_i^3), \\ \xi_i^2 & \text{if } (-\xi_i^1) = \xi_i^2 = +\xi_i^3, \\ 1 & \text{if } (-\xi_i^1) = \xi_i^2 = -\xi_i^3. \end{cases}$$

For example:

I^1	:	+	+	+	+	+	+	+	-	-	-	-	-	-
I^2	:	+	+	+	+	-	-	-	+	+	+	+	-	-
I^3	:	+	+	-	-	+	+	-	+	+	-	-	+	-
I^4	:	+	-	+	-	+	-	+	-	+	-	+	-	+
$I^{1.5}$:	+	+	+	+	+	+	+	+	+	+	+	-	-
$I^{1.75}$:	+	+	+	+	+	+	-	-	+	+	+	-	-

The interactions are given by

$$J_{ij} = \frac{1}{N} \left[\sum_{\mu=1}^P \xi_i^\mu \xi_j^\mu + g \sum_{\mu=1}^{P-1} \xi_i^{\mu+1} \xi_j^\mu \right].$$

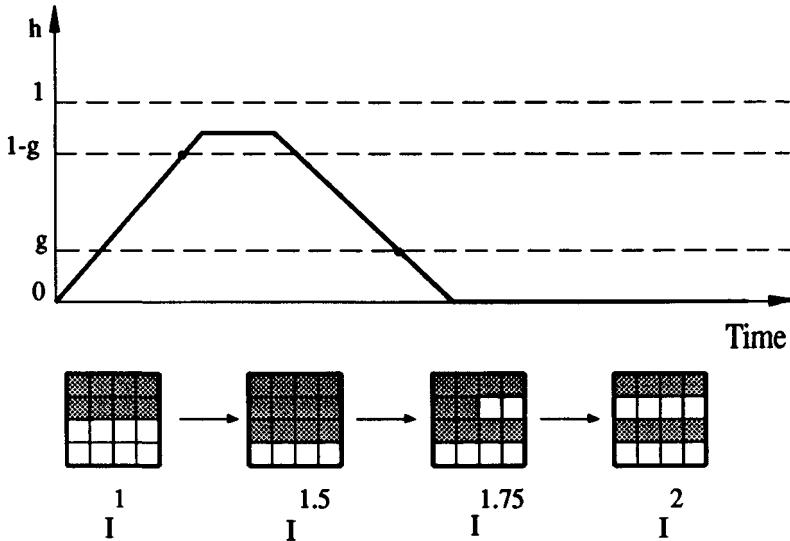


Figure 5.2. A temporal sequence driven by an external field h .

The dynamics is the following (see Fig. 5.2):

- i) One starts from I^1 .
- ii) The uniform field h is increased to a value $h_{\max} < 1$. I^1 is stable until $h = 1 - g$.
- Then the system jumps into state $I^{1.5}$: $I^{1.5}$ is stable as long as $h < 1$.
- iii) The field is decreased. The state $I^{1.5}$ is stable as long as $h > g$.
- iv) When $h < g$ the system jumps into a state which is the state $I^{1.75}$ provided that $\frac{3}{7} > g > \frac{1}{5}$.
- v) Then h is further decreased to zero and the system jumps into state I^2 .

Further details

When the system is in state I , the local field of a neuron i is given by

$$h_i(I) = \sum_{\mu=1}^{P-1} (\xi_i^\mu + g \xi_i^{\mu+1}) M^\mu + h.$$

- For state I^1 the field is $h_i(I^1) = \xi_i^1 + g\xi_i^2 + h$, ($M^1 = 1, M^\mu = 0, \mu \neq 1$). Then

either $\xi_i^1 = \xi_i^2$ and $h_i = (1+g)\xi_i^1 + h$

or $\xi_i^1 = -\xi_i^2$ and $h_i = (1-g)\xi_i^1 + h$.

Therefore the field h_i aligns along ξ_i^1 as long as $h < 1-g$. When the field is larger, all neurons with states $\xi_i^1 = -\xi_i^2$ align along h and I^1 becomes $I^{1.5}$. Then the overlaps are $M^1 = 0.5, M^2 = 0.5, M^\mu = 0, \mu \neq 1, 2$ and the local field becomes

$$h_i(I^{1.5}) = \frac{1}{2}\xi_i^1 + \left(\frac{1}{2} + \frac{1}{2}g\right)\xi_i^2 + \frac{1}{2}g\xi_i^3 + h.$$

- If $\xi_i^1 = \xi_i^2$ the local field $h_i(I^{1.5}) = \left(1 + \frac{1}{2}g\right)\xi_i^2 + \frac{1}{2}g\xi_i^3 + h$ is aligned along ξ_i^2 as long as $h < 1$, whatever the sign of ξ_i^3 .

If $\xi_i^1 = -\xi_i^2$ the local field is given by

$$h_i(I^{1.5}) = \frac{1}{2}g\xi_i^2 + \frac{1}{2}g\xi_i^3 + h.$$

If $\xi_i^2 = \xi_i^3$ the unit i aligns along ξ_i^2 , at least for $h < g$.

If $\xi_i^2 = -\xi_i^3$ the unit i is polarized along h .

Therefore if the field h is decreased from $1-g$ to g the state $I^{1.5}$ tends to transform into $I^{1.75}$. If, at this stage, the state $I^{1.75}$ is stable, a further decrease of h to zero makes certain that the state I of the system is driven towards I^2 since the overlap between $I^{1.75}$ and I^2 is large enough for the former state to be in the basin of attraction of the latter.

- But is $I^{1.75}$ a stable state? In state $I^{1.75}$ the overlaps are given by

$$M^1 = 0.25, \quad M^2 = 0.75, \quad M^3 = 0.25, \quad M^\mu = 0, \quad \mu \neq 1, 2, 3,$$

and the local field by

$$h_i(I^{1.75}) = \frac{1}{4}\xi_i^1 + \left(\frac{3}{4} + \frac{1}{4}g\right)\xi_i^2 + \left(\frac{1}{4} + \frac{3}{4}g\right)\xi_i^3 + \frac{1}{4}g\xi_i^4 + h.$$

If $\xi_i^2 = \xi_i^1 = \xi_i^3$ the field is $h_i = \left(\frac{5}{4} + g\right)\xi_i^2 + \frac{1}{4}g\xi_i^4 + h$, which must align along ξ_i^2 for $h = g$. The worst case is $\xi_i^4 = -\xi_i^2$ and therefore $\frac{5}{4} + \frac{3}{4}g > g$, whence $g < 5$.

Similarly, if $\xi_i^2 = \xi_i^1 = -\xi_i^3$ the field, which also must align along ξ_i^2 , is given by $h_i = \left(\frac{3}{4} - \frac{1}{2}g\right)\xi_i^2 + \frac{1}{4}g\xi_i^4 + h$, which leads to the following inequality:

$$\frac{3}{4} - \frac{3}{4}g > g, \quad \text{whence } g < \frac{3}{7}.$$

Likewise, the field associated with $\xi_i^2 = -\xi_i^1 = \xi_i^3$ must be parallel to ξ_i^2 . It is given by $h_i = \left(\frac{3}{4} + g\right)\xi_i^2 + \frac{1}{4}g\xi_i^4 + h$, which leads to $g < 3$.

Finally, the field with $\xi_i^2 = -\xi_i^1 = -\xi_i^3$ must be parallel to h :

$$h_i = \left(\frac{1}{4} - \frac{1}{2}g\right)\xi_i^2 + \frac{1}{4}g\xi_i^4 + h,$$

which yields $\frac{1}{4} - \frac{4}{4}g < g$, whence $g > \frac{1}{5}$.

The convenient range for g is therefore $\frac{3}{7} > g > \frac{1}{5}$.

- If h is reduced afterwards to zero all local fields h_i point towards the direction of ξ_i^2 and $I^{1.75}$ transforms into I^2 .

This system has been successfully simulated by Lehmann at zero noise level on networks comprising 2000 neurons. The sequence was made of 10 patterns. The best g he found was $g \simeq 0.25$.

5.2.3 Recall through delays

In this approach the skewed part of interactions J_{ij}^a is assumed to be conveyed through delays, whereas the symmetrical part is instantaneous (Amit, Sompolinsky *et al.*, Gutfreund). One has

$$J_{ij}^s = \frac{1}{N} \sum_{\mu} \xi_i^{\mu} \xi_j^{\mu}, \quad J_{ij}^a = \frac{g}{N} \sum_{\mu} \xi_i^{\mu+1} \xi_j^{\mu}. \quad (5.2)$$

For the sake of simplicity we assume that there is one delay ν_D . According to section 3.2.3 we are necessarily considering large networks and $\nu_D \gg \nu_0 = 1$. The local field is given by

$$\begin{aligned} h_i(\nu) &= \sum_j \left(\frac{1}{N} \sum_{\mu} \xi_i^{\mu} \xi_j^{\mu} \right) \sigma_j(\nu) + \sum_j \left(\frac{g}{N} \sum_{\mu} \xi_i^{\mu+1} \xi_j^{\mu} \right) \sigma_j(\nu - \nu_D) \\ &= \sum_{\mu} \xi_i^{\mu} M^{\mu}(\nu) + g \sum_{\mu} \xi_i^{\mu+1} M^{\mu}(\nu - \nu_D). \end{aligned}$$

Let us assume that from time 0 to ν_D the initial conditions are given by

$$M^1(\nu) = 1, \quad M^{\mu}(\nu) = 0, \quad \mu \neq 1, \quad \nu = 0, 1, \dots, \nu_D.$$

Then at time $\nu_D + 1$ the field becomes

$$h_i(\nu_D + 1) = \xi_i^1 + g \xi_i^2, \quad [h_i(\nu < \nu_D) = \xi_i^1],$$

and, provided that $g > 1$, the neuronal states flip from ξ_i^1 to ξ_i^2 . At time $\nu_D + 2$ the field is

$$h_i(\nu_D + 2) = (1 + g) \xi_i^2,$$

meaning that the state I^2 is very stable. When $\nu = 2\nu_D + 1$ the field becomes

$$h_i(2\nu_D + 1) = \xi_i^2 + g \xi_i^3.$$

The neuronal states σ_i becomes aligned along ξ_i^3 and so on (Fig. 5.3). This dynamics is robust. Instead of a single delay, a delay function $r(\nu)$ can be introduced:

$$h_i^a(\nu) = \sum_j J_{ij}^a \sum_{\nu'} r(\nu - \nu') \sigma_j(\nu').$$

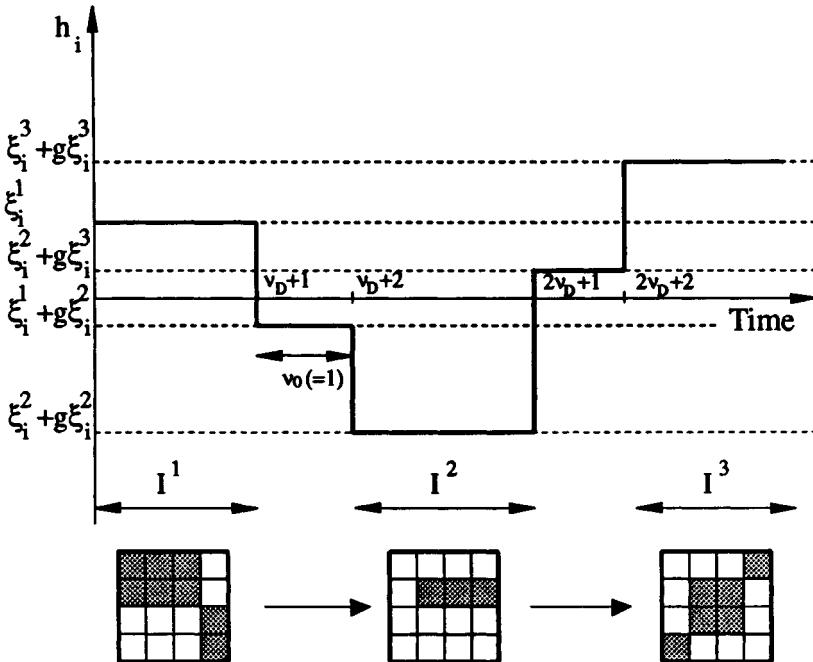


Figure 5.3. Recall of sequences through delays.

This system can also be used as a sequence recognizer. One fixes g to a value which is slightly less than 1 and an external sequence of patterns is applied to the network through fields:

$$h_i^{\text{ext}}(\nu) = K\xi_i^\mu, \quad (1 - \mu)\nu_D < \nu < \mu\nu_D, \quad K > 1 - g.$$

If the external sequence matches the internal one, the internal sequence is generated (Kleinfeld), otherwise it is not.

5.2.4 Synaptic plasticity

In this approach (Peretto, Niez) the synaptic efficacies are made of two parts:

- a) A static part which somewhat generalizes the Hopfield rule. It is written as

$$J_{ij}^L = \sum_{\mu\mu'} \xi_i^\mu \Gamma_{\mu\mu'} \xi_j^{\mu'}. \quad (5.3)$$

For example,

$$\Gamma = \begin{pmatrix} a & 0 & 0 & b \\ b & a & 0 & 0 \\ 0 & b & a & 0 \\ 0 & 0 & b & a \end{pmatrix}.$$

b) A plastic, reversible contribution J_{ij}^s which obeys the following equation:

$$\frac{dJ_{ij}^s}{dt} = -\frac{1}{\tau_a} \langle \sigma_i \sigma_j \rangle - \frac{J_{ij}^s}{\tau_s}. \quad (5.4)$$

The time constants are of the order of 10 to $30\tau_r$. Variations of synaptic efficacies on such time scales are all the more plausible. For example, ‘fast synapses’ have been proposed by Von der Malsburg. It must be stressed that the modifications of plastic efficacies depend only on the current activities of neurons.

Let us assume that the system is in state I^1 . The modifiable efficacies evolve so as to cancel out the contribution Γ_{11} for the static efficacies. The system ‘antilearns’ the pattern I^1 (owing to the minus sign in the dynamical equation of J_{ij}^s). The matrix Γ tends to become

$$\Gamma = \begin{pmatrix} 0 & 0 & 0 & b \\ b & a & 0 & 0 \\ 0 & b & a & 0 \\ 0 & 0 & b & a \end{pmatrix}.$$

The state I^1 is destabilized and, thanks to the off-diagonal term $\Gamma_{21} = b$, it jumps into the state I^2 . Then the state I^2 starts to destabilize, whereas the state I^1 becomes restabilized owing to the forgetting term of the dynamics of J_s , and Γ becomes:

$$\Gamma = \begin{pmatrix} a & 0 & 0 & b \\ b & 0 & 0 & 0 \\ 0 & b & a & 0 \\ 0 & 0 & b & a \end{pmatrix},$$

and so on (Fig. 5.4).

It is possible to generalize this algorithm by controlling every term of the matrix Γ . One simply states that if $\sigma_i(t) = \xi_i^\mu$ and $\sigma_j(t) = \xi_j^\mu$ then the coefficients must vary in such a way as to favor or to hinder the transition from I^μ to $I^{\mu'}$. This can be written as

$$\frac{dJ_{ij}^s}{dt} = +\frac{1}{\tau_a} \sum_{\mu\mu'} \Gamma_{\mu\mu'} \xi_i^\mu \xi_j^{\mu'} \left[\lambda_{\mu\mu'} \langle (\sigma_i(t) \xi_i^\mu)(\sigma_j(t) \xi_j^\mu) \rangle \right] - \frac{J_{ij}^s}{\tau_s},$$

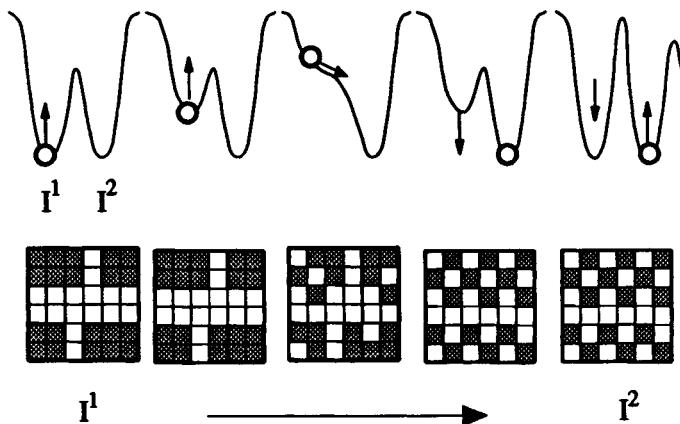


Figure 5.4. The principle of temporal sequences retrieval driven by fast evolving synaptic efficacies.

with $\lambda_{\mu\mu'} > 0$ if one wants to favor the transition $I^\mu \mapsto I^{\mu'}$, and $\lambda_{\mu\mu'} < 0$ otherwise. This equation can be rewritten as

$$\frac{dJ_{ij}^s}{dt} = \frac{1}{\tau_a^j} \langle \sigma_i \sigma_j \rangle - \frac{J_{ij}^s}{\tau_s},$$

with

$$\frac{1}{\tau_a^j} = \sum_{\mu\mu'} \Gamma_{\mu\mu'} \xi_j^\mu \xi_j^{\mu'} \lambda_{\mu\mu'},$$

a time constant which depends on the sites.

Computer simulations show that the algorithm is very efficient. It has even the surprising property of being self-correcting: sometimes the sequence is not perfectly retrieved when the process starts, but after a while the sequence arranges itself and the system provides faultless strings of patterns (see Figs. 5.5 and 5.6). The maximum frequency for pattern retrieval is of the order of $10\nu_r \simeq 50$ ms, which is compatible with the observations.

5.2.5 Using ternary synapses

This model (Dehaene *et al.*) makes use of heterosynaptic junctions.

A binary synapse (ij) is made of two terms:

- a) A constant inhibitory contribution $J_{ij}^L = -J^L$ which tends to destabilize any pattern.

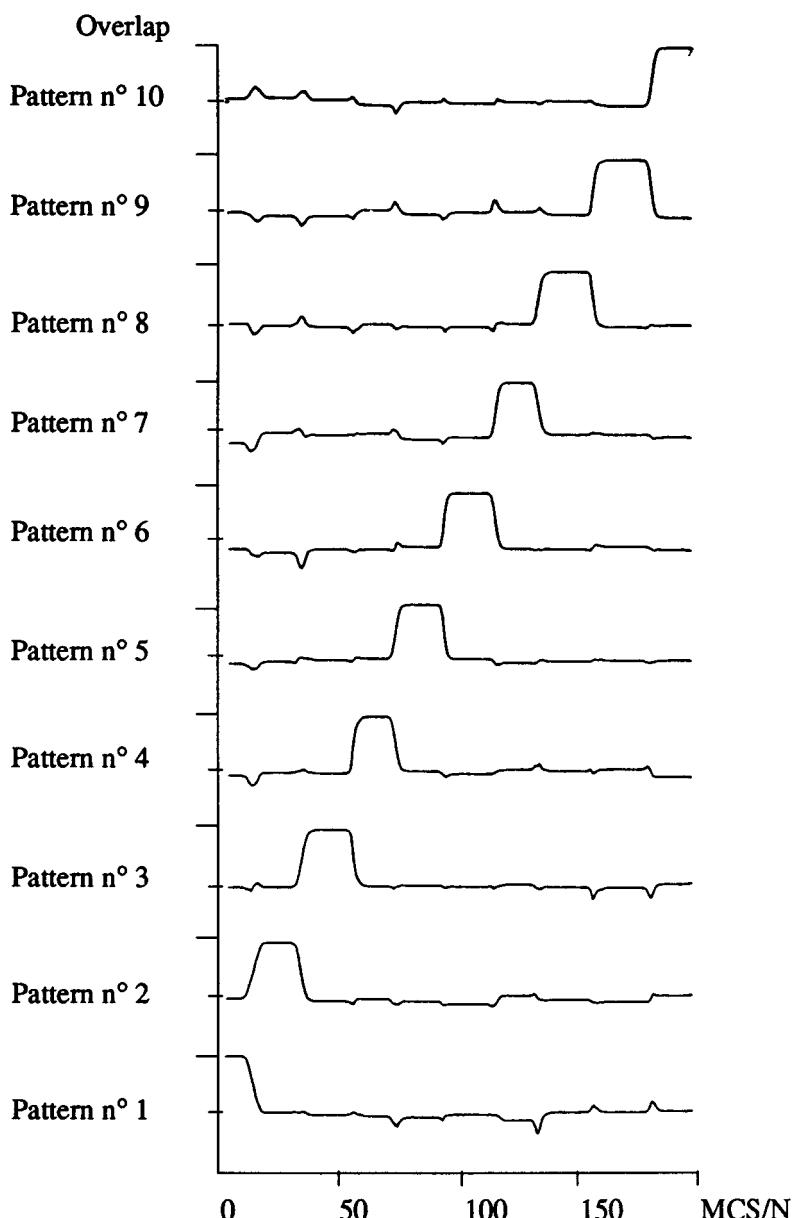


Figure 5.5. An example of a recall of a temporal sequence of 10 patterns in a 400-unit neural network according to the fast synaptic plasticity mechanism. The figure shows the evolution of the overlaps between the running state and the memorized patterns. The parameter b/a is of the order of 0.3. In this instance the sequence is perfect right from start.

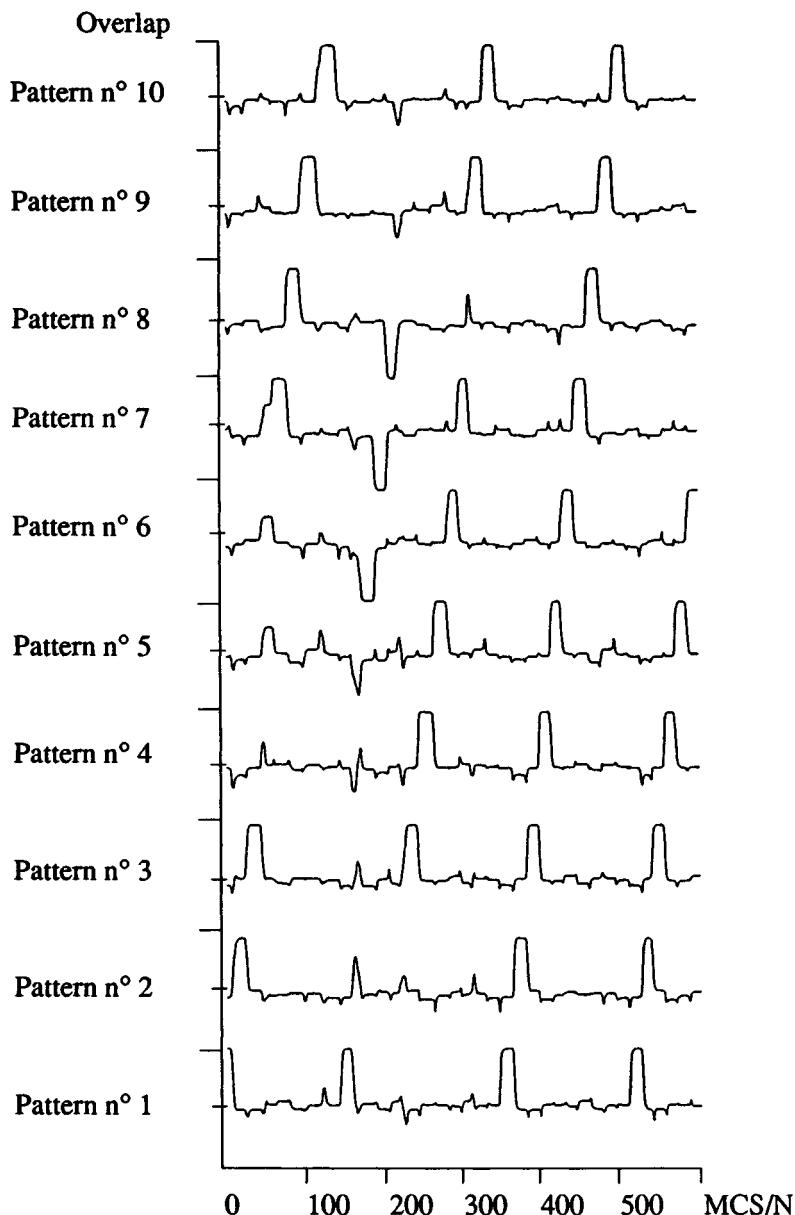


Figure 5.6. In this example the activity shows some incoherence when the network starts recalling the sequence of ten patterns. However, after a while, the synaptic dynamics seem to correct themselves and the right order of the sequence is recovered.

b) A plastic contribution $J_{ij(k)}^s$ which is modified through a heterosynaptic junction (ijk) according to the equation

$$\frac{dJ_{ij(k)}^s}{dt} = -\frac{J_{ij(k)}^s}{\tau_s} + \frac{1}{\tau_s} J_{ikj}(t) \langle \sigma_k(t) \rangle. \quad (5.5)$$

There is at most one connection k for every binary synapse (ij) . $(ij(k))$ labels the neuron k attached to a given synapse (ij) (see Fig. 5.7). One

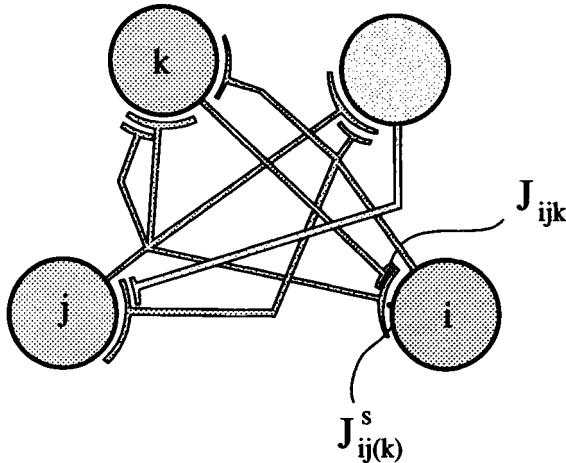


Figure 5.7. The model of Dehaene *et al.* appeals to ternary synapses J_{ijk} to account for the retrieval of temporal sequences of patterns. The efficacy of the binary synapse $J_{ij(k)}$ is modified by the activity of neuron k through the ternary synapse.

sees that the dynamics is not of the Hebbian (associative) type, since it depends only on the activity of one neuron, i.e. neuron k . Similarly the ternary synapse itself evolves and its dynamics is given by

$$\frac{dJ_{ijk}}{dt} = -\frac{J_{ijk}}{\tau'_s} + \frac{1}{\tau_t} \langle \sigma_i(t) \rangle. \quad (5.6)$$

It also depends on one activity, that of neuron i . One assumes that the time constants of the dynamics driving the evolution of the ternary synapse J_{ijk} are smaller than those driving the evolution of the binary synapse. Therefore

$$J_{ijk} \simeq \frac{\tau'_s}{\tau_t} \langle \sigma_i(t) \rangle$$

and

$$\frac{dJ_{ij(k)}^s}{dt} = -\frac{J_{ij(k)}^s}{\tau_s} + \frac{1}{\tau_b} \langle \sigma_i(t) \rangle \langle \sigma_k(t) \rangle.$$

One observes that the dynamics of the binary synapse is of the Hebbian type but the correlations it involves are not those between the activities of the neurons i and j that the synapse links. The mechanism is clear. Let us assume that the state of the network is I^1 . The dynamics builds synaptic strengths which correspond to a fixed point I^2 , the successor of I^1 . The two patterns are related by

$$\xi_{j(k)}^2 = \xi_k^1,$$

and, more generally,

$$\xi_{j(k)}^{\mu+1} = \xi_k^\mu.$$

The patterns of a sequence necessarily mirror the way the ternary synapses are distributed throughout the network, since only N^2 out of N^3 possible synapses are implemented in the system before any learning. The sequences are therefore random sequences. However, the system can learn: external sequences are generated and fed into the system through fields h_i^{ext} . The activities are computed by using the equations

$$\langle \sigma_i \rangle = S \left[\beta \left(\sum_j J_{ij}^L \sigma_j + \sum_{j(k)} J_{ij}^s \sigma_j + h_i^{\text{ext}} \right) \right]$$

and the learning dynamics is applied. If the strength of a ternary synapse dwindle to zero, it is eliminated.

This sort of Darwinian selection has indeed been observed at the neuronal muscular junction, for example (Changeux) (see section 9.2.4). The pruning of the ternary synapses selects certain sequences. This mechanism has been suggested as a possible model for the crystallization of birdsong. In actual fact the neurons of the model are not real neurons: the authors suggest that their properties are rather those of pools of neurons (the microcolumns?). One can therefore say that this is a model, probably the first, which takes the ultra-structure of the neuronal tissue into account.

5.2.6 Pacemakers. Habituation and threshold plasticity

Biological systems often need generators providing rhythmic, unlearnt patterns of activities. They can be created by special devices, i.e. special neurons as in the control of the cardiac rhythm, or they can be provided by ordinary neurons. The problem with ordinary neurons is that the only rhythm they are able to produce by themselves is the maximum firing rate.

a) One way out is to appeal to pools of neurons (Sompolinsky *et al.*). Let us first consider two neurons with antagonistic interactions:

$$J_{12} = -J_{21} = K.$$

We know that the asymptotic behavior of such an antisymmetrical system are limit cycles of length four. Indeed, the zero noise parallel dynamics generate the following sequence of states:

$$(++) \leftrightarrow (+-) \leftrightarrow (--) \leftrightarrow (-+) \leftrightarrow (++) \leftrightarrow \dots$$

Let us replace the two neurons by two pools of neurons. Every neuron of one pool is connected to every neuron of the other pool through antagonistic connections. The neurons of the first pool are all connected through excitatory connections J and the neurons of the second pool have no connections between themselves. The dynamics of this system is given by the usual equations:

$$\begin{aligned}\frac{dM^1}{dt} &= -\frac{1}{\tau} [M^1 - S(\beta(JM^1 - KM^2))], \\ \frac{dM^2}{dt} &= -\frac{1}{\tau} [M^2 - S(\beta KM^1)],\end{aligned}$$

where M^1 and M^2 are the average activities of the two pools.

It must be reminded that the form of these equations does not depend on the type of dynamics. Their study shows that for large enough K s and low enough noises, the system oscillates. The oscillations are relaxation oscillations and their frequency depends on the parameters J/K and β .

b) The pacemaker we just described appeals to pools of neurons. However, many pacemakers one observes in natural systems are made of a few, well-identified neurons.

Let us consider a simple neuron and let it be excited by an external field h (a constant depolarizing potential).

$$\langle \sigma_1 \rangle = S(\beta(h - \theta)).$$

One observes that the neuron first fires at its maximum rate and then the firing rate decays. This is habituation, a phenomenon which can be simply modeled by assuming that the threshold depends on the activity of the neuron:

$$\theta = \theta(\langle \sigma(t) \rangle).$$

Habituation is the key of a simple model of a pacemaker made of two mutually inhibiting neurons. The activities of the two neurons are

given by

$$\begin{aligned}\frac{d\langle\sigma_1\rangle}{dt} &= -\frac{1}{\tau} \left[\langle\sigma_1\rangle - S(\beta(-K\langle\sigma_2\rangle - \theta_1(t))) \right], \\ \frac{d\langle\sigma_2\rangle}{dt} &= -\frac{1}{\tau} \left[\langle\sigma_2\rangle - S(\beta(-K\langle\sigma_1\rangle - \theta_2(t))) \right],\end{aligned}$$

with $K > 0$. If the thresholds are fixed, $\theta_1 = \theta_2 = 0$, the states $\sigma_1 = -\sigma_2 = 1$ or $\sigma_1 = -\sigma_2 = -1$ are stable and there is no oscillation. However, if one assumes that the threshold depends on the cell activities

$$\begin{aligned}\frac{d\theta_1}{dt} &= -\frac{\theta_1 - (a\langle\sigma_1\rangle + \theta_0)}{\tau_\theta}, \\ \frac{d\theta_2}{dt} &= -\frac{\theta_2 - (a\langle\sigma_2\rangle + \theta_0)}{\tau_\theta},\end{aligned}$$

with $a > 0$, the system oscillates and the pattern of activities which is observed in the central pattern generators (CPG) of crustacea, for example, is well reproduced. θ_0 is the threshold for $\langle\sigma\rangle = 0$.

The dynamics of large neural networks with dynamical thresholds has been studied by Horn and Usher. The dynamics of the thresholds is given by

$$\frac{d\theta_i}{dt} = -\frac{\theta_i - (a\langle\sigma_i\rangle + \theta_0)}{\tau_\theta}.$$

Horn *et al.* consider two cases:

a) $\theta_0 = 0$. — The authors observe that the system either oscillates between M^μ and $-M^\mu$ or that it relaxes towards a constant activity. If a pointer, an asymmetrical contribution to interactions (5.1), is introduced, the state jumps from one state towards its successor, but this desired behavior is spoiled by frequent sign reversals.

b) $\theta_0 = a$. — This condition amounts to saying that the threshold is modified only by active neurons. Its value increases exactly as in the habituation phenomenon. Horn *et al.* then observe that the system jumps according to the pointer and that there are no more sign reversals of order parameters.

5.3 An example of conditioned behavior

Gelperin and Hopfield have put forward a model of classical conditioning for the tastes in an invertebrate, the slug. The animal has a series of chemical sensors, a ganglion which is supposed to store the various combinations characterizing the different sorts of food, carrot, beetroot, . . . ,

offered to the slug and a genetically determined circuit which makes the animal flee when it tries noxious food (quinine) (see Fig. 5.8).

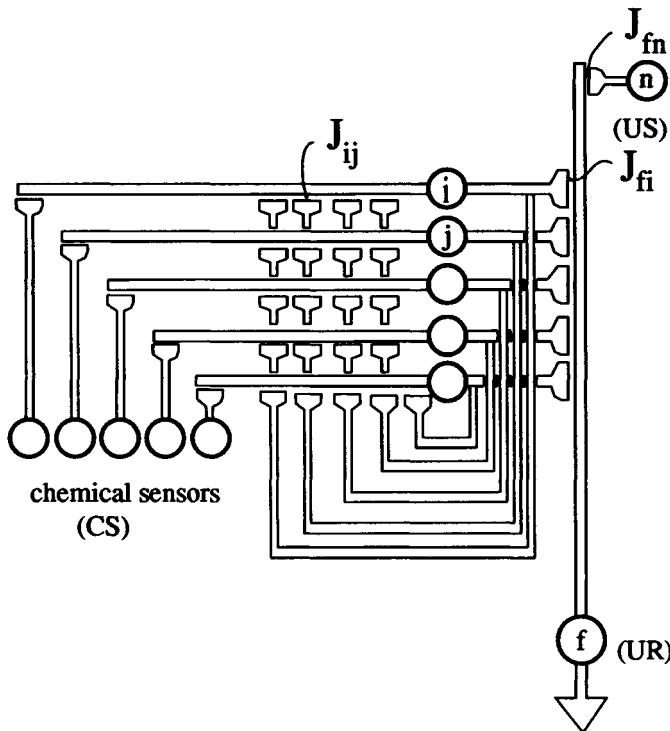


Figure 5.8. A schematic network which models the olfactory system of a slug (After Gelperin and Hopfield).

Let $I^\mu = \{\xi_i^\mu = \pm 1\}$, $\mu = 1, \dots, P$, $i = 1, \dots, N$ be the combinations of sensory inputs associated with a certain food μ . i labels the neurons of the associative net. The patterns are stored in the associative net using a Hebbian rule,

$$J_{ij} = \frac{1}{N} \sum_{\mu=1}^P \xi_i^\mu \xi_j^\mu. \quad (5.7)$$

Various sorts of carrots corresponding to slightly disturbed versions of the archetypal 'carrot' elicit the same response of the associative net. On the other hand, an excitation of the nociceptive sensor always triggers

the response of the fleeing system, which is modeled by a neuron f . The outputs of the neurons i and that of the nociceptive sensor make contact with f . The synapses (f_i) are modifiable according to a Hebbian rule:

$$\Delta J_{fi} = \varepsilon \sigma_i \sigma_f, \quad \varepsilon > 0. \quad (5.8)$$

When a food μ is associated with a noxious excitation the synapses are reinforced till the food μ alone triggers the fleeing response. In this classical conditioning experiment the nociceptive signal is the US, the fleeing behavior is the UR and the food μ is the CS.

Several remarks are to be made:

a) The described experiment and simulations are examples of a negative reinforcement. Positive reinforcement has also been considered by the authors who added another circuit comprising a sensor triggering positive responses such as absorbing the food. The system now classifies three sorts of food, noxious, attractive and neutral.

b) Instead of putting binary synapses into play the model could have relied upon heterosynaptic junctions as well. Let n be the nociceptive neuron. Since if $\sigma_n = 1, \sigma_f = 1$ in all circumstances the learning rule (3.108) could have been written as

$$\Delta J_{in(f)} = \varepsilon \sigma_i \sigma_n.$$

Mathematically this substitution makes no difference, but it is biologically important. It means that the plasticity does not depend on the activity of f , but that it depends on the coactivation of the axons of i and those of n on the dendrites of f . This heterosynaptic effect has been observed.

c) The model is not really a good model, since it does not account for the temporal relationship between the US and the CS, which one knows is necessary for the conditioning to be efficient. The authors then add inhibitory axo-axonal contacts between the output of f and the outputs of neurons i . The excitation of f blocks the incoming of signals emitted by the neurons and association is possible only if the CS signal precedes the US signal as it must be.

d) The explanation of classical conditioning could rely more heavily on collective behaviors:

Let us consider a fully connected network. Some connections are genetically determined. These connections create a number of attractors I^{μ_u} in the phase space which correspond to stereotyped behaviors (or an unconditioned response US^{μ_u}).

$$J_{ij}^u = \frac{1}{N} \sum_{\mu_u=1}^{P_u} \xi_i^{\mu_u} \xi_j^{\mu_u}.$$

An excitation (US^{μ_u}) brings the system to the corresponding state I^{μ_u} . On the other hand, the system learns other patterns I^{μ_s} corresponding to conditioned stimuli (CS^{μ_s}):

$$J_{ij}^s = \frac{1}{N} \sum_{\mu_s=1}^{P_s} \xi_i^{\mu_s} \xi_j^{\mu_s}.$$

Association couples the two types of attractors I^{μ_s} with $I^{\mu_{u(s)}}$:

$$J_{ij}^{As} = \frac{1}{N} \sum_{\mu_s} \xi_i^{\mu_{u(s)}} \xi_j^{\mu_s}.$$

Let us start with a conditioned stimulus (CS^{μ_s}). The system settles in the associated basin of attraction I^{μ_s} and, after a while, one of the mechanisms used to trigger the recall of temporal sequences is put into play, driving the system from I^{μ_s} to $I^{\mu_{u(s)}}$. Obviously the order the two patterns are associated in matters. If I^{μ_s} precedes $I^{\mu_{u(s)}}$ the off-diagonal elements J_{ij}^{As} makes I^{μ_s} more unstable vis-à-vis $I^{\mu_{u(s)}}$ and conditioning is inefficient.

THE PROBLEM OF LEARNING IN NEURAL NETWORKS

6.1 Introducing the problem

6.1.1 Beyond Hebbian models

The Hebbian form of synaptic efficacies is inspired by classical conditioning experiments and by the associativity paradigm which these observations suggest. The learning dynamics of Hebbian networks is given by

$$J_{ij}(\mu) = J_{ij}(\mu - 1) + \varepsilon \xi_i^\mu \xi_j^\mu, \quad \varepsilon > 0, \quad (6.1)$$

and the iteration may stop once all P patterns have been experienced by the network, so giving the Hebbian rule of Eq. (4.2). We have seen that the performances of such Hebbian models are unfortunately limited: the maximum number of information bits they are able to store is of the order of $\alpha_c N^2$, with $\alpha_c \ll 1$, whereas the number of bits which is embedded in synaptic efficacies is larger than N^2 . Hebbian networks are therefore not that efficient at storing information and it is not surprising that the learning dynamics given by Eq. (6.1) is not optimal, which amounts to saying that there exist sets of parameters J_{ij} which improve the performances. This leads to the following questions:

- a) What is the maximum memory storage capacity of a neural network whose architecture is given? Then,
- b) How are we to obtain the parameters, synaptic efficacies and thresholds, of optimal networks?

Partial answers such as the memory storage capacity of most simple networks are known. Also, new learning dynamics which allow the maximum capacity to be reached have been found. These topics are discussed in this chapter and in Chapter 7.

6.1.2 The phase space of interactions

In Chapter 3 the study of the state dynamics of a neural network led us to introduce an N -dimensional space, the phase space of the network. A point I of the phase space represents a state of the net. Likewise it is convenient to define an $N(N + 1)$ -dimensional space to describe the

dynamics of interactions. The number of dimensions is the sum of the number of synaptic efficacies J_{ij} ($\neq J_{ji}$ in general), which is N^2 , and of the number of thresholds, which is N . A point \mathbf{J} of this space fully represents a particular realization of the network. \mathbf{J} is a solution to the problem of learning if the system behaves as expected, for example if its fixed points correspond to the patterns to be memorized.

It must be realized that learning is an ill-defined problem. Either the problem may have no solution or it may accept a whole set of solutions \mathbf{J} . In the latter case the set determines a *volume Γ of solutions*.

The problem of learning therefore is two-fold:

- Decide whether the problem has at least one solution. If the problem has no solution it is pointless to train the network to find one. This is to be related to question a) of section 6.1.1
- If the volume Γ of solutions is non-zero, devise a learning algorithm which brings the vector \mathbf{J} inside Γ . This is to be related to question b).

It is certainly very useful to have analytical results regarding Γ . For example the limit capacities of the network are obtained by studying the shrinking of Γ to zero. The size of Γ also gives indications on how easy is the problem: the larger Γ , the easier it is to find a route (a learning dynamics) that brings \mathbf{J} into Γ . For a given network, Γ is specific to the problem \mathcal{P} which is to be solved. It may be useful to associate an entropy $S(\mathcal{P})$ to \mathcal{P} . This entropy is defined by

$$S(\mathcal{P}) = \log(\Gamma(\mathcal{P})).$$

Only a few analytical results have been obtained so far on the computation of volumes of solutions. The two main contributions, which are discussed in the following sections, are due to Cover on the one hand (see section 6.2.3) and to Gardner on the other (in section 6.3). The use by E. Gardner of the techniques of statistical mechanics to carry out this computation paves the way to well-founded theories of learning.

6.1.3 Neural networks and Boolean mappings

The essence of classical conditioning is to associate stimuli with responses. The stimuli are coded on a number of N_I *input units*. The responses are coded on a number of N_O *output units*. Input and output units, taken together, make the set of *visible units*. In Hebbian networks all neurons are visible, which means that the overall number N of neurons is given by

$$N = N_I + N_O.$$

A pattern $I^\mu = I^{\mu, \text{in}} \otimes I^{\mu, \text{out}}$ is made of

- an input state $I^{\mu, \text{in}} = \{\xi_{j \in \mathcal{I}}^{\mu, \text{in}}\}, \quad j = 1, \dots, N_I, \quad \mu = 1, \dots, P;$
- an output state $I^{\mu, \text{out}} = \{\xi_{i \in \mathcal{O}}^{\mu, \text{out}}\}, \quad i = 1, \dots, N_O.$

As shown by Eq. (6.1), in Hebbian training no distinction is made between input and output units during the learning session. The distinction takes place during the retrieval phase, where it is necessary to specify the units which materialize the stimuli and those which materialize the responses.[†]

Hebbian networks, as well as more general neural nets, can therefore be considered as systems which transform an input signal into an output signal. Since the neural states are one-bit states, a neural network realizes a Boolean mapping between a set of N_I binary states and a set of N_O binary states. A learning session has been successfully carried out if the network yields the correct mapping for the whole set of P patterns. In general the number P of patterns is less than the number \mathcal{N}_I of possible input states. This number is

$$\mathcal{N}_I = 2^{N_I}.$$

When $P = \mathcal{N}_I$ the mapping is called a Boolean function. The number of possible output states is

$$\mathcal{N}_O = 2^{N_O}.$$

Since there are \mathcal{N}_O possible output states for every input state, the number \mathcal{N}_B of possible Boolean functions is

$$\mathcal{N}_B = (\mathcal{N}_O)^{\mathcal{N}_I} = (2^{N_O})^{2^{N_I}},$$

a number that gets tremendously large even for small networks.

This leads to a second series of questions. Namely:

- c) What sort of Boolean mappings (or function) can be implemented in a neural network whose architecture is given?
- d) How are we to build a neural network which implements a given Boolean mapping (or function)?

It must be noted that an input state is related to an unique output state, and therefore it is necessary that $N_O \leq N_I$. In particular, the

[†] Here arises the question of determining the ratio N_O/N_I of the number of output units relative to that of input units. We want these ratios to be as large as possible, but it is obvious that too large ratios hinder the retrieval properties of the network. The limit is determined by the size R of basins of attraction. Crude estimates of R may be obtained. In particular we have seen in section 4.1.4 that the size of basins shrinks to zero when the loading parameter $\alpha = P/N$ becomes critical $\alpha \rightarrow \alpha_c$. It must be stressed that no analytical treatment of the sizes of basins in fully connected Hebbian networks has been carried out so far.

number of N -input, one-output Boolean functions (categorizer systems) is given by

$$\mathcal{N}_B = 2^{2^N}.$$

A neural network which is made of N_I input units and one output unit is called *a perceptron* (see Fig. 6.1). This is the most simple architecture one can imagine. Naturally it has been the focus of extensive researches. The main results on perceptrons are explained below.

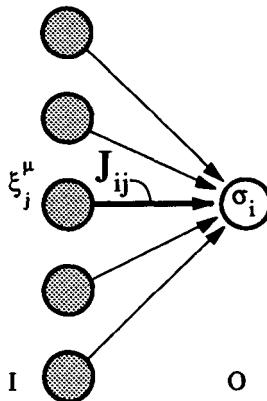


Figure 6.1. The perceptron architecture.

The abilities of the perceptron architecture are quite limited, and implementing Boolean mappings demands, in general, the introduction of more complicated architectures. In particular, it is necessary to consider a new category of neurons, the so-called *hidden units* (see Fig. 6.2). Hidden units are neurons whose states are determined neither by the stimuli nor by the responses while the system is trained.

To summarize, the neurons are labeled along the following scheme:

$$N \text{ units} = \begin{cases} N_V & \text{visible units} \\ N_I & \text{input units} \\ N_O & \text{output units} \\ N_H & \text{hidden units.} \end{cases}$$

The problem of choosing which states the hidden units should take during training sessions is central in the theory of learning. It is called the *problem of credit assignment*. Some solutions are given in Chapter 8.

6.1.4 Implementing general Boolean functions in neural networks by using logic-minded solutions

We look for an N -input, one-output neural network which associates

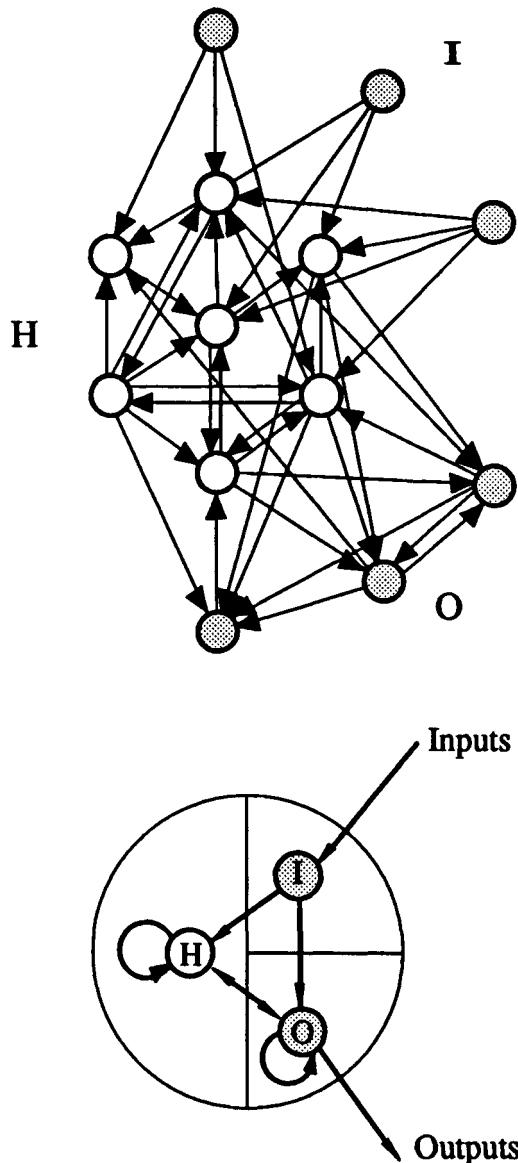


Figure 6.2. Classification of neural units.

I and O are visible units.

I: Input; O: output; H: hidden units.

the 2^N possible input binary patterns:

$$I^{\mu, \text{in}} = (\xi_j^{\mu, \text{in}} \in \{+1, -1\}), \quad \mu = 1, \dots, 2^N, \quad j = 1, \dots, N,$$

each with one of the two output states:

$$\xi^{\mu, \text{out}} \in \{+1, -1\}.$$

The network possibly comprises hidden units. There are P patterns I^μ , which yields a *positive* $\xi^{\mu, \text{out}} = +1$ output state. All other $2^N - P$ patterns I^μ are associated with a negative output $\xi^{\mu, \text{out}} = -1$. Figure 6.3

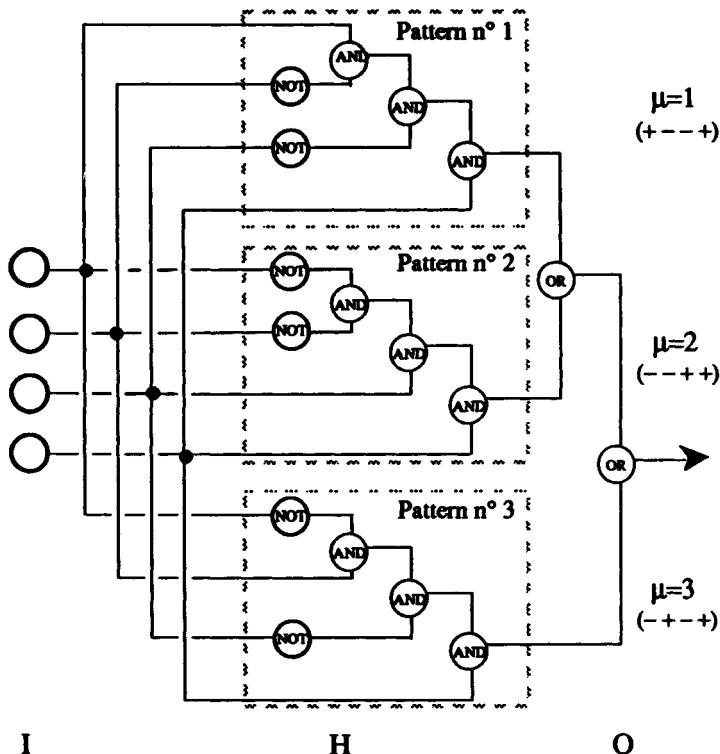


Figure 6.3. A logic-minded neural network which yields $\xi^{\text{out}} = +1$ for the patterns defined in Eq. (6.2) and $\xi^{\text{out}} = -1$ otherwise.

gives an example with $N = 4$ (16 possible input states) and $P = 3$. The associations are the following:

$$\begin{array}{ccccccc}
 +1 & -1 & -1 & +1 & \leftrightarrow & +1 \\
 -1 & -1 & +1 & +1 & \leftrightarrow & +1 \\
 -1 & +1 & -1 & +1 & \leftrightarrow & +1.
 \end{array} \tag{6.2}$$

Simple solutions can be imagined. The problem is that they are very neuron-demanding. The first is a *logic-minded network*. The net is built by first feeding P processing units with the N input lines. If the j th bit of the μ th pattern is $\xi_j^\mu = -1$, the corresponding line has a NOT gate so as to make all outputs of the μ th processor equal to +1 if the system experiences the pattern I^μ .

Then a cascade of OR gates, one for every processor, check whether all incoming signals are in states +1, which is the sign that the system experiences the pattern associated with that processor. Finally, a cascade of AND gates indicates whether one of the OR cascades is active. This system carries out the correct associations. We saw in Chapter 1 how to make NOT, AND and OR gates by using neural units. We therefore have succeeded in the building of a neural network that is able to implement any Boolean function. But the price to pay is high. On average one needs $\frac{1}{2}N \times P$ NOT gates, $(N-1) \times P$ AND gates and P OR gates. The number of hidden neurons is therefore

$$N_H = \frac{3}{2}NP.$$

For general Boolean functions the number of positive outputs is $P \simeq 2^{N-1}$ and the number of hidden units grows exponentially with the number of input states.

6.1.5 The grandmother cells solution

The characteristic property of a *grandmother cell* is that it responds to one and only to one specific input state (the grandmother face for example). The *grandmother cells solution* consists in building a hidden layer of P units with the property that a hidden unit, $j = j(\mu)$, is active if and only if the input state is a specific state $I^{\mu, \text{in}}$ (Fig. 6.4). P has the same meaning as in section 6.1.4. The internal representation of the network can be viewed as a $P \times P$ table:

	internal neurons $j \dots$
label	+ - - - - - - - -
of	- + - - - - - - -
patterns	- - + - - - - - -
μ	.. .
:	- - - - - - - - - +

The state of the hidden unit j is given by

$$\sigma_j^{\text{hid}}(I^\mu) = \begin{cases} +1 & \text{if } j = j(\mu), \\ -1 & \text{if } j \neq j(\mu). \end{cases}$$

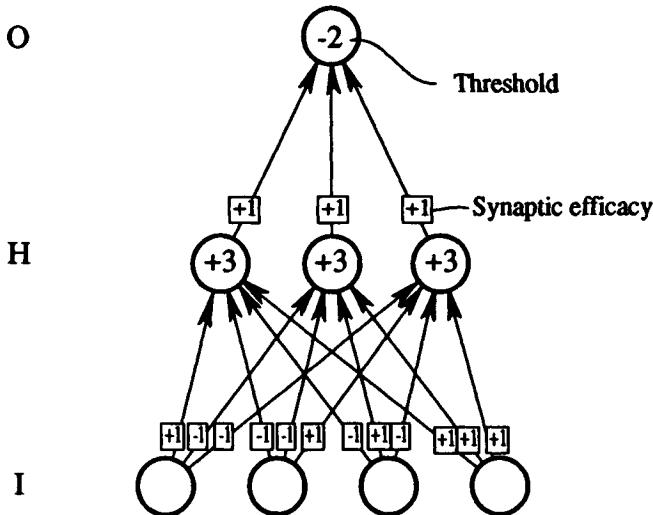


Figure 6.4. A grandmother neural network for the Boolean function (6.2).

For this to be true it is enough to choose

$$J_{j \in \mathcal{H}, k \in \mathcal{J}} = \xi_k^{\mu, \text{in}}, \quad J_{j, 0} = -\theta_j = -N + 1,$$

where μ is the pattern such as $j = j(\mu)$, since

$$\sum_{k \in \mathcal{I}} J_{jk} \xi_k^{\mu} = \begin{cases} N & \text{for } j = j(\mu), \\ \sum_{k \in \mathcal{I}} J \leq N - 2 & \text{otherwise.} \end{cases}$$

The mapping between the hidden layer and the output unit is easy. The connections are given by

$$J_{\text{out}, j \in \mathcal{H}} = +1.$$

For a given input pattern $I^{\mu, \text{in}}$ the field h^{out} on the output unit is

$$\begin{aligned} h^{\text{out}}(I^{\mu}) &= \sum_{j \in \mathcal{H}} J_{\text{out}, j} \sigma_j^{\text{hid}}(I^{\mu}) + J_{\text{out}, 0} \\ &= 1 - (P - 1) = 2 - P + J_{\text{out}, 0} \end{aligned}$$

for all patterns, such as $\xi^{\text{out}} = +1$ and

$$h^{\text{out}}(I^{\mu}) = -P + J_{\text{out}, 0}$$

for all other patterns, which shows that the convenient threshold on the output unit is:

$$-\theta_{\text{out}} = J_{\text{out},0} = P - 1.$$

The grandmother solution for the Boolean function (6.2) is given in Fig. 6.4. If the training set is the whole set of input states the number P of hidden units is 2^{N-1} . As in the logic-minded solution the size of the internal layer increases exponentially with the size of the input space. Nevertheless, there is an improvement in the number of units by a factor of $3N$ which may be substantial for lower values of P .

6.1.6 On the complexity of neuronal architectures

There exist many solutions to the problem of finding a neural network that implements a given Boolean function. Let us consider for example the XOR function. This is a two-input, one-output function which is defined by

$$\begin{array}{ccccc} + & + & \mapsto & - \\ + & - & \mapsto & + \\ - & + & \mapsto & + \\ - & - & \mapsto & - \end{array}$$

Figure 6.5 displays solutions involving tri-layered networks with either one or two or three (or more) units in the hidden layer.

A grandmother solution of the XOR function only needs two hidden neurons but, as one can see, there exists a more compact network.

Any neural net, even the simplest, realizes a Boolean function between its input units and its output units. Let us then consider the Boolean function that is generated by a simple neural network. We assume that we know the function but not the network itself and we want to build a system that realizes the mapping. Appealing to logic-minded or grandmother techniques is certainly not economical, since the number of units of the so built networks is of the order of 2^N , whereas the number of units of the generating network may be much lower, for example it may be of the order of N . In actual fact, one knows that the problem of finding the minimal set of logical functions that implements a given Boolean function is a hard problem. It is even the archetype of a whole class of hard problems which are called NP-complete problems. Finding the minimal neural network, that is to say finding the network with the smallest number of units, is therefore also a NP-complete problem.

In general it is enough to find a ‘good’ solution which is a network whose number of units is close to, though larger than, that of the optimal network (the best solution). Let ϵ be the difference. To devise algorithms which find good solutions it is necessary that the number of networks

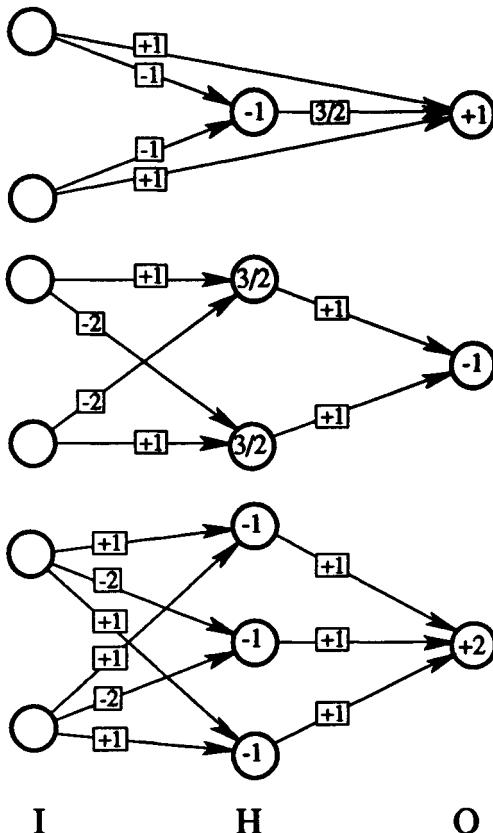


Figure 6.5. How to make an XOR function using a hidden layer of one, two or three units.

that are ε -optimal increases exponentially with ε . If the case arises good networks comprise a polynomial, not an exponential, number of neurons. We still ignore how the number of neural nets which solve the problem of Boolean mappings increases with ε but there exists an encouraging result, namely the Kolmogorov theorem:

Given any continuous function $f(x_1, x_2, \dots, x_N)$ defined for all points of an N -dimensional hypercube, $\{x_k\} \in [-1, +1]^N$, there exist $N(2N+1)$ continuous sigmoidal functions $h_{jk}(x_k)$ and one continuous (generally non sigmoidal) function $g(y)$, so that

$$f(x_1, x_2, \dots, x_N) = \sum_j^{2N+1} g\left(\sum_k^N h_{jk}(x_k)\right).$$

This formula looks very similar to that given by a tri-layered neural network with N input units, $2N + 1$ hidden units and a single linear output unit. It embeds genuine Boolean functions just as well, since it is not forbidden to assume that $f \in \{-1, +1\}$ at the vertices $\{x_k\}$ in $\{-1, +1\}^N$ of the hypercube. The problem is, first, that the theorem gives no clue regarding the way the functions can be found and, second, that the function $g(x)$ may be very complex, not at all resembling a neuronal response function. In actual fact the whole complexity of the N -dimensional landscape determined by $f(\{x_k\})$ is packed in $g(y)$. Nevertheless, the Kolmogorov theorem raises the hope that not-too-complicated functions g , those of neuronal types in particular, could lead to acceptable approximations of f . This is why the question of learning in layered neural networks is the focus of intensive research.

We have seen that the problem of finding the minimal network which implements a given Boolean function is NP-complete. The converse problem is that of determining the set of Boolean functions that can be implemented in a neural network whose architecture is given. Judd *et al.* have shown that this is also a NP-complete problem.

6.2 Linear separability

6.2.1 Linearly separable Boolean functions

The structure of a perceptron is fully determined by the set of its synaptic efficacies J_j , $j = 1, \dots, N$, where N is the number of units of the input layer and $-J_0 = \theta$, the threshold of the unique output unit (see Fig. 6.1). The index of the target neuron may be removed from the label of synaptic efficacies, since there is no ambiguity. A given perceptron determines a well-defined Boolean function \mathcal{B} of all possible input patterns $I^{\mu, \text{in}} = \{\xi_j^{\mu, \text{in}}\}$, with $\xi_j^{\mu, \text{in}} \in \{-1, +1\}$, on the two possible states of the output unit:

$$\sigma^{\mu, \text{out}} = \mathcal{B}(\{\xi_j^{\mu, \text{in}}\}) = \text{sign}\left(\sum_{j=0}^N J_j \xi_j^{\mu, \text{in}}\right). \quad (6.3)$$

The Boolean functions which can be implemented in a simple N -input one-output perceptron form a subset of all possible Boolean functions. They are called *linearly separable functions*.

The XOR Boolean function, for example, cannot be implemented in an $N = 2$ perceptron, which amounts to saying that the Boolean function XOR is non-linearly separable. This is easy to understand: the field acting on the output unit is given by

$$h(I^{\mu, \text{in}}) = J_1 \xi_1(I^{\mu, \text{in}}) + J_2 \xi_2(I^{\mu, \text{in}}) + J_0.$$

This is the equation of a straight line in the (ξ_1, ξ_2) plane whose parameters are J_1 , J_2 and J_0 . The problem is finding a set of parameters which brings apart the points $(+1, +1)$ and $(-1, -1)$ from the points $(+1, -1)$ and $(-1, +1)$ (see Fig. 6.6). There does not exist such a set and the XOR function cannot be implemented in the perceptron.

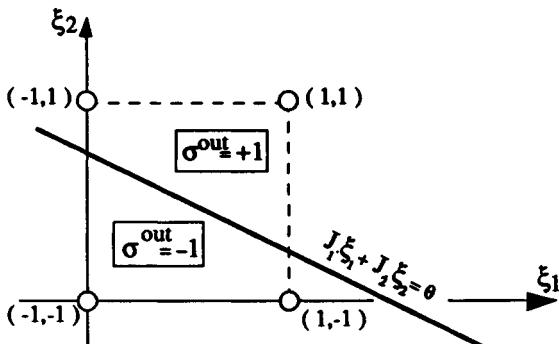


Figure 6.6. The non-separability of the XOR problem. It is not possible to find a line which drives the opposite corners of the square apart.

Let $\mathcal{N}_L(N)$ be the number of linearly separable Boolean functions with N arguments. The problem of determining \mathcal{N}_L is solved in section 6.2.3. \mathcal{N}_L is given approximately by

$$\mathcal{N}_L \simeq 2 \frac{2^{N^2}}{N!}. \quad (6.4)$$

When compared with the total number $\mathcal{N}_B = 2^{2^N}$ of Boolean functions, \mathcal{N}_L dwindles to zero as the number N of input units increases.

One says that the set of linearly separable functions is of zero measure. This discovery, which had been put forward by Minsky and Pappert, shows that the perceptron is unable to trap even simple logical relationships. It was felt as a definite failure of the perceptron.

Table 6.1 makes conspicuous how fast a simple perceptron becomes unable to implement most Boolean mappings already for $N = 3$ (in column 4 a factor of 2 has been taken into account according to the results of Eq. (6.8)).

6.2.2 On the number of linearly separable functions

The set of the 2^N possible input states I^{in} can be viewed as the 2^N vertices of an N -dimensional hypercube. The result of a given Boolean function $B(I^{in})$ applied to an input I^{in} is a + or - output, a sign which can be attached to the vertex I^{in} of the

N	Boolean	L. sep. (exact)	Up. bd. (Eq. (6.9))	Eq. (6.4)
1	4	4	4	4
2	16	14	18	16
3	256	104	130	170
4	65536	?	3394	5460
5	4294967296	?	411478	559240

Table 6.1. 2nd col.: number of N -input Boolean functions;
 3rd col.: number of linearly separable functions;
 4th col.: upper bound given by Eq. (6.9);
 5th col.: approximation given by Eq. (6.4).

hypercube. Therefore a set of + or - signs associated with all vertices determines a Boolean function. A Boolean function is linearly separable if there exists at least one $(N - 1)$ -dimensional hyperplane which drives apart all vertices with a 'plus' sign from all vertices with a 'minus' sign.

We now consider the N -dimensional space spanned by the interaction vectors \tilde{J} . Equation (6.3) shows that an interaction vector \tilde{J} supplemented with a threshold $-J_0$ fully determines one Boolean function. According to whether the sign of J_0 is positive or negative, Eq. (6.3) may be rewritten as

$$\sigma^{\text{out}} = \text{sign} \left(\sum_{j=1}^N \frac{J_j}{|J_0|} \xi_j^{\text{in}} + \frac{J_0}{|J_0|} \right) \begin{cases} = \text{sign} \left(\sum_{j=1}^N J_j \xi_j^{\text{in}} + 1 \right) \\ = \text{sign} \left(\sum_{j=1}^N J_j \xi_j^{\text{in}} - 1 \right) \end{cases}$$

where the rescaling of synaptic efficacies $J_j/|J_0| \mapsto J_j$ has been carried out. Therefore a (rescaled) vector \tilde{J} determines two Boolean functions, say $B_{\tilde{J}}^+$ and $B_{\tilde{J}}^-$.

When the vector \tilde{J} goes through one of the 2^N hyperplanes \mathcal{P}^+ defined by

$$\sum_{j=1}^N J_j \xi_j^{\mu, \text{in}} + 1 = 0, \quad (6.5)$$

for example through the plane associated with $I^{\mu_0, \text{in}} = \{\xi_j^{\text{in}, \mu_0}\}$, the result of the Boolean function $B_{\tilde{J}}^+(I^{\text{in}, \mu_0})$ on the input I^{in, μ_0} changes. Therefore a Boolean function is associated with every volume determined by the set of hyperplanes \mathcal{P}^+ . Let \mathcal{N}_V be the number of volumes. A similar reasoning shows that \mathcal{N}_V Boolean functions $B_{\tilde{J}}^-$ are associated with the volumes determined by the 2^N hyperplanes \mathcal{P}^- defined by

$$\sum_{j=1}^N J_j \xi_j^{\text{in}, \mu} - 1 = 0. \quad (6.6)$$

The construction of the $2 \times \mathcal{N}_V$ volumes generates all possible linearly separable Boolean functions. This is therefore an upper bound for the number \mathcal{N}_L of separable functions:

$$\mathcal{N}_L \leq 2\mathcal{N}_V. \quad (6.7)$$

Two Boolean functions B_j^+ corresponding to two different volumes cannot be identical. Likewise, the Boolean functions B_j^- are different. Moreover, since the number of +1 outputs for the functions B_j^+ is larger than or equal to $\frac{1}{2}2^N$, and since this number is lower than or equal to $\frac{1}{2}2^N$ for the functions B_j^- , the set of functions B_j^+ is disconnected from the set of functions B_j^- , except for a few cases corresponding to functions, which we call balanced Boolean functions, yielding an equal number of +1 and -1 outputs. Let N_{eq} be the number of volumes associated with the balanced functions. Symmetry considerations show that the Boolean functions B_j^+ and B_{-j}^- are identical in these volumes. Therefore the number of linearly separable functions is

$$N_L = 2N_V - N_{\text{eq}}.$$

We now give an upper bound for N_V . There is one \mathcal{P}^+ hyperplane associated with every vertex of the hypercube that describes the input states. Two hyperplanes which correspond to opposite vertices are parallel to each other. The problem is therefore to find the number N_{par} of volumes determined by 2^{N-1} pairs of parallel hyperplanes. This number is an upper bound of N_V , since it does not take into account the decrease in the number N_{deg} of volumes brought about by degenerate intersections of hyperplanes. N_L is finally given by

$$N_L = 2(N_{\text{par}} - N_{\text{deg}}) - N_{\text{eq}}. \quad (6.8)$$

Let T_P^N be the number of volumes generated by P pairs of parallel planes in an N -dimensional space. Following the line of reasoning we introduce in the next section (6.2.3), we find that

$$T_P^N = T_{P-1}^N + 2T_{P-1}^{N-1},$$

since the number of volumes created by two parallel hyperplanes in the $(N-1)$ -dimensional space is twice the number created by a single hyperplane. The solution of this equation is obtained by noting that each move brings a factor of 2 in the counting (see section 6.2.3). Therefore T_P^N is given by

$$\begin{aligned} T_P^N &= \sum_{k=0}^N 2^k \binom{P}{k} \\ \text{and } N_{\text{par}} &= T_{2N-1}^N = \sum_{k=0}^N 2^k \binom{2^{N-1}}{k}. \end{aligned} \quad (6.9)$$

Let us study the number of linearly separable functions of $N = 3$ inputs. There are $2^3 = 8$ possible input states and the number of Boolean functions is $2^8 = 256$. The number of volumes given by Eq. (6.9) is $N_{\text{par}} = 65$. There exist 6 points where 4 planes meet. For example, the point $(0, 0, -1)$ belongs to the 4 planes:

$$\mp J_1 \mp J_2 + J_3 + 1 = 0.$$

This convergence removes one volume for each point and therefore $N_{\text{deg}} = 6$. Finally, one observes that six volumes with tetragonal symmetries and eight volumes with trigonal symmetries correspond to balanced Boolean functions. Then we have

$N_{\text{eq}} = 14$, and finally Eq. (6.8) gives the following number of linearly separable Boolean functions of three Boolean input states:

$$N_{\mathcal{L}} = 2(65 - 6) - 14 = 104.$$

We do not want to go into the analysis of N_{deg} , nor into that of N_{eq} . It is enough to know that these quantities are marginal as compared with N_{par} to derive an expression given the leading order of $N_{\mathcal{L}}$. For large enough N s all terms of Eq. (6.9) are negligible when compared with the last contribution,

$$N_{\mathcal{L}} \simeq 2 \times 2^N \binom{2^{N-1}}{N} \simeq \frac{2 \times 2^N (2^{N-1})^N}{N!} = 2 \frac{2^{N^2}}{N!}, \quad (6.10)$$

which is Eq. (6.4).

6.2.3 The Cover limit

A N -unit visible neural network may be considered as a set of perceptrons and the problem of learning in visible networks reduces to that of learning in perceptrons. We then, examine the implementation of *Boolean mappings* in perceptrons. A Boolean mapping is determined by a collection of P patterns:

$$I^\mu = I^{\mu, \text{in}} \quad (= \{\xi_i^{\mu, \text{in}}\}) \cup \xi^{\mu, \text{out}}, \quad i = 1, \dots, N, \quad \mu = 1, \dots, P \leq 2^N.$$

Owing to the limited abilities of perceptron architectures there exists an upper bound, P_c , to the number of patterns that can be mapped. It happens that the limit $\alpha_c = P_c/N$ is sensitive to data structures and in the following analysis one assumes that the learning set \mathcal{E} of patterns I^μ is made of *random patterns*. The limit α_c is then an absolute limit which has nothing to do with the learning rule that is used to realize the mapping. The comparison of this value with that which one finds by using a particular learning rule gives an estimate on how efficient the rule is.

A pattern $I^\mu = \{\xi_i^\mu\}$, $i = 1, 2, \dots, N + 1$ is stable if for all i s one has

$$x_i^\mu = \xi_i^\mu h_i(I^\mu) = \sum_{j \neq i} J_{ij} \xi_i^\mu \xi_j^\mu > 0, \quad (6.11)$$

where x_i^μ is the *stabilization parameter* of pattern I^μ on site i . We associate the following N -dimensional vectors $\tilde{\tau}_i^\mu$ and \tilde{J}_i with neuron i :

$$(\tilde{\tau}_i^\mu)_j = \xi_i^\mu \xi_j^\mu \quad \text{and} \quad (\tilde{J}_i)_j = J_{ij}, \quad i \neq j.$$

\tilde{J}_i is the bundle of connections impinging on neuron i . With this notation the equations (6.11) become:

$$x_i^\mu = \tilde{J}_i \cdot \tilde{\tau}_i^\mu > 0. \quad (6.12)$$

For example, to memorize three patterns in a 2-neuron network the following six inequalities must be simultaneously satisfied:

$$\begin{aligned}x_1^1 &= \tilde{J}_1 \cdot \tilde{\tau}_1^1 > 0 & x_2^1 &= \tilde{J}_2 \cdot \tilde{\tau}_2^1 > 0, \\x_1^2 &= \tilde{J}_1 \cdot \tilde{\tau}_1^2 > 0 & \text{and} & x_2^2 = \tilde{J}_2 \cdot \tilde{\tau}_2^2 > 0, \\x_1^3 &= \tilde{J}_1 \cdot \tilde{\tau}_1^3 > 0 & & x_2^3 = \tilde{J}_2 \cdot \tilde{\tau}_2^3 > 0.\end{aligned}$$

One observes that the problem of finding the $N = 3$ synaptic efficacies of synapses impinging on neuron 1 is decoupled from that of finding the three efficacies of synapses impinging on neuron 2. From the point of view of learning, the network may be considered as made of two disconnected perceptrons, as stressed above, and the question is reduced to looking for a vector \tilde{J} which simultaneously satisfies the P inequalities,

$$x^\mu = \tilde{J} \cdot \tilde{\tau}^\mu > 0, \quad (6.13)$$

an equation where the index i has been omitted.

The Cover theorem states that a perceptron, and therefore a neural network comprised of N visible neurons can store a maximum of $2 \times N$ random patterns in the limit of large N s.

Proof (by M. Griniasty). — In the N -dimensional space spanned by \tilde{J} , the P equations

$$\begin{aligned}x^1 &= \tilde{J} \cdot \tilde{\tau}^1 = 0, \\x^2 &= \tilde{J} \cdot \tilde{\tau}^2 = 0, \\&\dots \\x^P &= \tilde{J} \cdot \tilde{\tau}^P = 0,\end{aligned} \quad (6.14)$$

delimit a number R_P^N of regions. A string of P bits given by $\text{sign}(x^\mu)$ is associated with every region. Therefore for a given set of P patterns one finds R_P^N such strings, for example $(+ + - \dots)$, $(+ - + \dots)$, \dots . A string tells whether the individual inequalities are satisfied or not in a given region (see Fig. 6.7). If the set of strings does not comprise the string $(+ + \dots +)$ there does not exist a vector \tilde{J} satisfying the P inequalities (6.13). The number of possible strings of P bits is 2^P .

Since the patterns to be stabilized are chosen at random, every one of the R_P^N strings plays the same role and the probability of finding the string $(+ + \dots +)$ is given by

$$\bar{\omega}_R(N, P) = \frac{R_P^N}{2^P}.$$

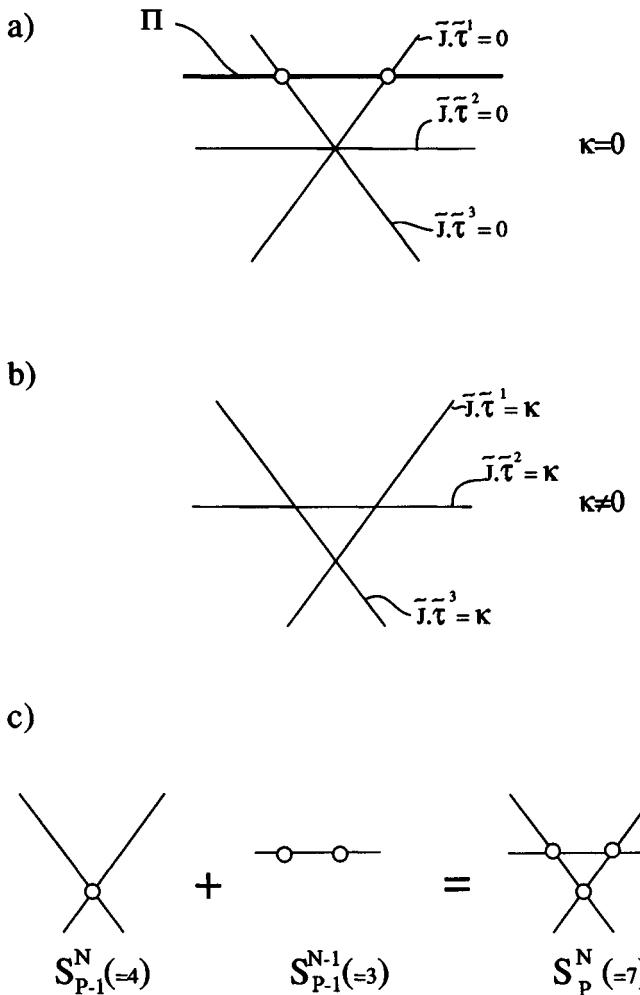


Figure 6.7. Counting the number of regions in the phase space of interactions (schematic).

- With no bias κ .
- With a non-zero bias κ .
- The iterative equation for the number of regions.

The storage capacity P_c is defined by

$$\bar{\omega}_R(N, P_c) = \frac{1}{2} \cdot$$

The stabilization parameter x^μ is a measure of how well imprinted (on a given neuron) is pattern I^μ . Satisfying the inequalities (6.13) makes sure

that I^μ is stable, but this stability can be very small. A way of increasing stability, that is of making the state I^μ less sensitive to perturbations, is to replace the conditions (6.13) by more stringent conditions:

$$x^\mu = \tilde{J} \cdot \tilde{\tau}^\mu > \kappa, \quad (6.15)$$

where $\kappa > 0$ is a bias. The P lines of \tilde{J} plane determined by the equations

$$\tilde{J} \cdot \tilde{\tau}^1 = \kappa, \quad \tilde{J} \cdot \tilde{\tau}^2 = \kappa, \dots, \tilde{J} \cdot \tilde{\tau}^P = \kappa,$$

delimit a number S_P^N of regions with $S_P^N > R_P^N$. It happens that the computation of S_P^N is easier than that of R_P^N . However, the quantity:

$$\bar{\omega}_S(N, P) = \frac{S_P^N}{2^P}$$

has no physical meaning, since the volume of solutions explicitly depends on the bias $\Gamma = \Gamma(\kappa)$ and the probability of finding a solution with non-zero biases cannot be calculated by using only combinatorial arguments. The actual computation of $\Gamma(\kappa)$ is carried out in section 6.3. However, in the limit $\kappa \rightarrow 0$, all volumes that are determined by the hyperplanes are identical and one expects that

$$\bar{\omega}_S(N, P) \simeq \bar{\omega}_R(N, P).$$

The proof is given below (Eq. 6.17). One then starts by computing $\bar{\omega}_S(N, P)$. S_P^N obeys a recursion relation:

$$S_P^N = S_{P-1}^N + S_{P-1}^{N-1}.$$

To prove the relation one observes that, starting with the S_{P-1}^N volumes delimited by $(P-1)$ conditions in the N -dimensional space, the addition of a new condition creates new volumes. There is a one-to-one correspondence between these new volumes and the volumes determined by the $(P-1)$ conditions in the $(N-1)$ -space (see Fig. 6.8).

The recursion relation also holds for the unbiased conditions:

$$R_P^N = R_{P-1}^N + R_{P-1}^{N-1}.$$

The solution of the recursion relation is

$$S_P^N = \sum_{k=0}^{\min(P, N)} \binom{P}{k}. \quad (6.16)$$

This can be showed by assigning a box of coordinates (P, N) to S_P^N in a P, N plane (see Fig. 6.8). The recursion relation states that S_P^N is

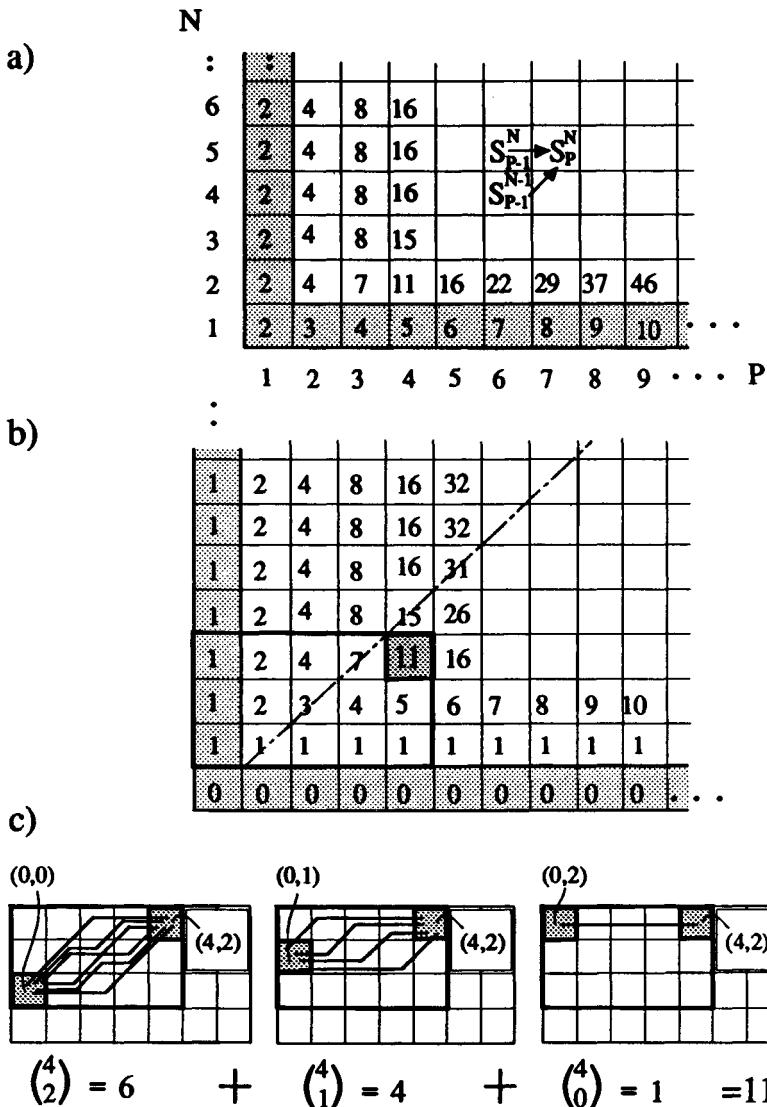


Figure 6.8. Solving the iterative equation.

the sum of the contents of its western and south-western boxes. The trick is to add one column and two rows to the table, showing that all the values of S_P^N can be computed starting from the boxes of the left column which contain only ones or zeros. One defines an allowed path

in the table as a path which propagates either eastwards (*e*) or north-eastwards (*n*) at every step. A given 1 in a box of the left column gives a contribution to the (P, N) box if there exists at least one allowed path between these two boxes. In actual fact the contribution of this 1 to the content S_P^N of the (P, N) box is the number of allowed paths, starting from the $(P = 0, N - k)$ box and arriving at the (P, N) box. Let us then consider this $(0, N - k)$ box of the left column. A string of P symbols (*e*) or (*n*) with k symbols (*n*) such as

$$e \ e \ n \ e \ n \ n \ e \ e \ n \ e \dots$$

determines an allowed path between the two boxes. There are as many paths as there are ways of combining k symbols among N symbols, that is $\binom{P}{k}$ paths. All the contributions arising from the various values of k must be added. We see from the table that the boxes of the first column which can contribute to S_P^N are those with $k \leq P$, and therefore

$$S_P^N = \sum_{k=0}^P \binom{P}{k}.$$

However, if $P > N$ there exist paths starting from boxes containing zero values which do not contribute to S_P^N . If the case arises the maximum value of k is $k = N$, whence formula (6.16).

We are now in the position of computing the number R_P^N of volumes determined by P unbiased conditions in an N -dimensional space. It is related to S_{P-1}^{N-1} , the number of volumes determined by $(P - 1)$ biased conditions in an $(N - 1)$ -dimensional space, by the relation

$$R_P^N = 2 S_{P-1}^{N-1}, \quad (6.17)$$

whence (Winder)

$$R_P^N = 2 \sum_{k=0}^{N-1} \binom{P-1}{k}, \quad P > N. \quad (6.18)$$

To prove the relation we consider an $(N - 1)$ -dimensional hyperplane Π , with Π parallel to one of P -planes defined by the P unbiased conditions (Fig. 6.7). The relation between R_P^N and S_{P-1}^{N-1} is made clear by realizing that, on the one hand, the volumes in Π are $(N - 1)$ -dimensional volumes determined by $(P - 1)$ biased conditions and, on the other, that one volume in Π is associated with exactly two (unbiased) volumes in the N -dimensional space.

In the limit of large values of N , the asymptotic expressions of $\bar{\omega}_S$ and $\bar{\omega}_R$, which are derived from equations (6.16) and (6.18), are the same.

The memory storage capacity of the perceptron can therefore be deduced from the study of $\bar{\omega}_S$.

a) If $P < N$, then $\min(P, N) = P$ and

$$S_P^N = \sum_{k=0}^P \binom{P}{k} = (1+1)^P = 2^P;$$

therefore

$$\bar{\omega}_S(N, P) = 1;$$

which means that there always exists a solution whatever the set of memorized patterns.

b) Let us look at the value $P = 2 \times N$. Then

$$\bar{\omega}_S(N, P) = \frac{1}{2^{2N}} \sum_{k=0}^N \binom{2N}{k},$$

$$\text{but } \sum_{k=0}^N \binom{2N}{k} = \sum_{k=N+1}^{2N} \binom{2N}{k} = \frac{1}{2} 2^{2N} \text{ and therefore}$$

$$\bar{\omega}_S(N, P) = \frac{1}{2}.$$

$\bar{\omega}_S(N, P)$ is a sigmoidal curve which always passes through the point ($\alpha = P/N = 2$; $\bar{\omega}_c = 0.5$). It tends towards a step function as the number of neurons increases to infinity (the thermodynamic limit). This means that the inequalities (6.13) can be verified on every site i as long as $P < 2 \times N$. For a full pattern to be stable it is necessary for the inequalities to be satisfied simultaneously on all sites. This stability can be achieved by large enough networks. The question of determining the minimal size is discussed in section 6.2.4.

6.2.4 Storing random patterns

The probability that P patterns can be perfectly stored is given by

$$(\bar{\omega}_S(N, P))^N.$$

We look for an analytical expression of

$$\bar{\omega}_S(N, P) = \sum_{k=0}^N \frac{1}{2^P} \binom{P}{k} = \sum_{k=0}^N Q(P, k), \quad P > N.$$

Using the Stirling formula $n! \simeq n^n e^{-n} \sqrt{2\pi n}$, $Q(P, k)$ can be written as

$$Q(P, k) = \left(\frac{1}{2} \pi P \right)^{-1/2} \exp \left[-\frac{2(k - \frac{1}{2}P)^2}{P} \right]$$

and

$$\bar{\omega}_S(N, P) = \left(\frac{1}{2} \pi P \right)^{-1/2} \int_0^N dk \exp \left[-\frac{2(k - \frac{1}{2}P)^2}{P} \right].$$

This expression is transformed

$$\bar{\omega}_S(N, P) = \frac{1}{\sqrt{\pi}} \int_{-\sqrt{P/2}}^{\sqrt{2/P}(N-P/2)} dt \exp(-t^2) \simeq \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\sqrt{2/P}(N-P/2)} dt \exp(-t^2),$$

and becomes

$$\begin{aligned} \bar{\omega}_S(N, P) &= \frac{1}{2} \left[\frac{2}{\sqrt{\pi}} \int_{-\infty}^0 dt \exp(-t^2) + \frac{2}{\sqrt{\pi}} \int_0^{\sqrt{2/P}(N-P/2)} dt \exp(-t^2) \right] \\ &= \frac{1}{2} \left(1 + \operatorname{erf} \left[\sqrt{2/P} \left(N - \frac{1}{2} P \right) \right] \right). \end{aligned}$$

Using the expansion of the error function, $\operatorname{erf}(x) \simeq 1 - \frac{1}{x\sqrt{\pi}} \exp(-x^2)$, it takes the following form:

$$\bar{\omega}_S(N, P) \simeq 1 - \frac{2}{\epsilon \sqrt{\pi N}} \exp \left(-\frac{1}{4} N \epsilon^2 \right),$$

with $\alpha = P/N \simeq 2$ and $\epsilon = 2 - P/N = 2 - \alpha$.

The global probability of imprinting all patterns without any mistake is given by

$$\log(\bar{\omega}_S^N) = N \log(\bar{\omega}_S) \simeq -\frac{2}{\epsilon} \sqrt{\frac{N}{\pi}} \exp \left(-\frac{1}{4} N \epsilon^2 \right).$$

One sees that this probability is mainly determined by the exponent. The size N of a network which perfectly stores $P = \alpha N$ patterns diverges as

$$N \simeq \frac{4}{(2 - \alpha)^2}.$$

More precisely, to store for example $P = 1.9 \times N$ patterns with one chance in two that the storage will proceed without any error, one must appeal to networks comprising at least N neurons, with N given by

$$N = 200 \times \log N \simeq 1116,$$

one finds $N \simeq 2696$, a large network indeed.

6.2.5 Orthogonal patterns

The limit capacity $P_c = 2 \times N$ is rather surprising. Let us go back to our analysis of Hebbian models. We recall that the Hebbian efficacies are given by

$$J_{ij} = \frac{1}{N} \sum_{\mu=1}^P \xi_i^\mu \xi_j^\mu$$

and that the local field $h_i(I^{\mu_0})$ can be decomposed into a signal term h_i^s (a coherent contribution according to the terminology of section 4.1.3)

$$h_i^s(I^{\mu_0}) = \xi_i^{\mu_0},$$

and a noise, or incoherent, local field h_i^n :

$$h_i^n(I^{\mu_0}) = \frac{1}{N} \sum_{\mu \neq \mu_0}^P \sum_j^N \xi_i^\mu \xi_j^\mu \xi_j^{\mu_0}.$$

The pattern I^{μ_0} is stable if the coherent stability parameter x_i^{s,μ_0} ,

$$x_i^{s,\mu_0} = \xi_i^{\mu_0} h_i^s = 1$$

overcomes the incoherent stability parameter x_i^{n,μ_0} ,

$$x_i^{n,\mu_0} = \frac{1}{N} \sum_{\mu \neq \mu_0}^P \sum_j^N \xi_i^\mu \xi_i^{\mu_0} \xi_j^\mu \xi_j^{\mu_0} = \sum_{\mu \neq \mu_0}^P \xi_i^\mu \xi_i^{\mu_0} (I^\mu \cdot I^{\mu_0}).$$

The best one can do to store a maximum of patterns is to cancel out all scalar products $I^\mu \cdot I^{\mu_0}$ which appear in the noise term. Therefore the maximum capacity is related to the number of patterns that can be made orthogonal in an N -unit network.

The maximum number P_c of strings of N bits which can be made simultaneously orthogonal is N .

Proof. — Strictly speaking, this property is valid when N is a power of 2: $N = 2^r$.

The components ξ_i^μ of a pattern I^μ can be considered as special values of components of general vectors V in an N -dimensional space. One knows that the maximum number of mutually orthogonal vectors one can build in this space is N and therefore $P_c \leq N$.

On the other hand it is possible to build N orthogonal sets of N bits. As the construction eventually misses other possibilities, one has $P_c \geq N$. Therefore $P_c = N$.

The construction is as follows:

- To one state (+) one associates two states (++) and (+-) of length 2. These two states are orthogonal:

$$+ \mapsto \begin{cases} ++ \\ +- \end{cases}$$

- Each of the two states is considered as one new state $\dot{+}$ on which the same production rule is applied. For example,

$$\dot{+} (\equiv++) \mapsto \begin{cases} \dot{++} (\equiv++++) \\ \dot{+-} (\equiv++--) \end{cases} \quad \text{and} \quad \dot{+} (\equiv+-) \mapsto \begin{cases} \dot{+-} (\equiv+-+-) \\ \dot{--} (\equiv---+). \end{cases}$$

- The orthogonality holds for the states built on $\dot{+}$ and $\dot{-}$ as it holds for the states built on + and -. The four states we constructed are orthogonal. The process can then be pursued:

$$+ \mapsto ++ (\equiv \dot{+}) \mapsto \dot{++} (\equiv \ddot{+}) \mapsto \ddot{++} (\equiv \ddot{\dot{+}}) \mapsto \dots$$

It generates 2^τ orthogonal patterns of N bits ($N = 2^\tau$) after τ steps.

One extra pattern, I^{N+1} , cannot be made orthogonal to the first N ones. The best that can be achieved is to choose this pattern such as

$$I^{N+1} \cdot I^\mu = \frac{1}{\sqrt{N}}.$$

Then the noise term is of the order of 1 and it can overcome the signal term.

The conclusion is that $P = N$ is the limit of memory storage capacities for orthogonal patterns. We insist on the fact that this value of $\alpha_c = 1$ is a consequence of the orthogonality of memorized patterns. Randomizing the patterns to be memorized increases the capacity to $\alpha_c = 2$ (the Cover limit). Further increasing the correlations between the patterns still enhances the capacity, which may tend to infinity in the limit of very large overlaps (as the case arises in sparse coding). This point is examined in section 7.3. Conversely, making the patterns more orthogonal results in a decreasing of the maximum memory storage capacity.

6.3 Computing the volume of solutions

6.3.1 A direct scrutiny into the space of interactions

The calculation by Cover of the maximum storage capacity of neural networks is interesting, but it is difficult to have it generalized to more involved cases. For example, we would like to determine how many random patterns can be stored with a stability parameter $x^\mu > \kappa$. Or it would be useful to assess the effects of inter-pattern correlations on the storage capacities of the networks. E. Gardner has proposed a versatile technique which allows the carrying out of such generalizations. The principle of the technique is the following:

A pattern I^μ is stable if $\tilde{J} \cdot \tilde{\tau}^\mu > 0$. Therefore a vector \tilde{J} is a solution of the memorization problem if

$$\prod_{\mu=1}^P \mathbf{1}(\tilde{J} \cdot \tilde{\tau}^\mu) = 1.$$

The volume of solutions is given by

$$\Gamma = \sum_{\tilde{J}} \prod_{\mu} \mathbf{1}(\tilde{J} \cdot \tilde{\tau}^\mu), \quad (6.19)$$

where \tilde{J} is normalized, $|\tilde{J}|^2 = N$. The step function $\mathbf{1}(x)$ is exponentiated:

$$\mathbf{1}(x) = \int_0^\infty d\lambda \delta(\lambda - x) = \int_0^\infty \frac{d\lambda}{2\pi} \int_{-\infty}^{+\infty} dy \exp(iy(\lambda - x)), \quad (6.20)$$

where the Fourier transform of the Dirac function $\delta(x)$ has been used. When the expression (6.20) is introduced into Eq. (6.19) one observes that the calculation of Γ is similar to the calculation of a partition function with an ‘energy’ given by the exponent of (6.20). It is then possible to appeal to the tools of statistical mechanics to find Γ . We explain below in detail how the technique is applied to the problem of finding the number of patterns that one can store with stability parameters larger than a given minimum value of κ .

The calculation yields

$$\alpha_c(\kappa) = \frac{P_c(\kappa)}{N} = \frac{1}{\int_{-\kappa}^{+\infty} \frac{dy}{\sqrt{2\pi}} (\kappa + y)^2 \exp(-\frac{1}{2}y^2)}. \quad (6.21)$$

In the limit $\kappa \rightarrow 0$ this equation gives $\alpha_c = 2$, in agreement with the results of Cover. The behavior of the limiting capacity when the bias κ increases is depicted in Fig. 6.9. It must be stressed that this curve represents the maximum number of patterns which a perceptron can store with a stability larger than the bias κ . This is not the number of stable patterns which is $P_c = 2 \times N$ whatever the bias κ . The additional information which is provided by the Gardner computation is the distribution of the depths of basins of attraction of memorized patterns. This distribution $\rho(\kappa)$ is given by $\rho(\kappa) = -d\alpha_c(\kappa)/2d\kappa$. If, moreover, one assumes that the deeper the basin the larger it is, the calculation also gives an idea of the distribution of the sizes of basins.

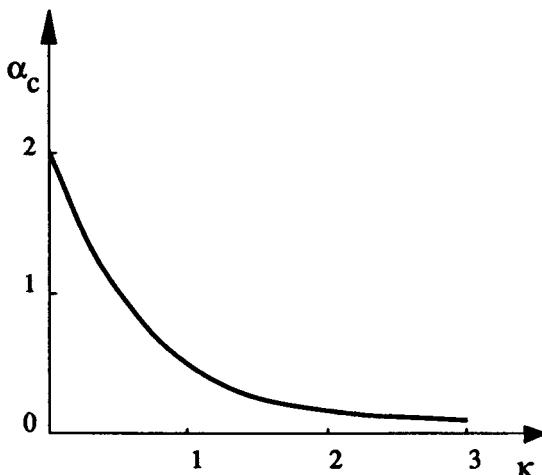


Figure 6.9. Memory storage capacity as a function of the bias κ . For $\kappa = 0$ the capacity is $\alpha_c = 2$, the Cover limit (After Gardner).

The Gardner approach is extremely interesting because it yields information on the distribution of interactions which solve a given type of problem. It tells how large the volume Γ of available solutions is. To find the convenient algorithm, one so devised that it brings the interactions right in this target volume, is another problem.

6.3.2 Detailing the calculation

This part is a little technical and in spite of its interest we think it is better to skip it on first reading.

A set of interactions $\{J_j\}$ of a perceptron stabilizes the patterns I^μ :

$$I^\mu = \xi_j^\mu \otimes \xi^{\mu, \text{out}}, \quad \xi_j^\mu, \xi^{\mu, \text{out}} \in \{+1, -1\}, \quad j = 1, \dots, N, \mu = 1, \dots, P,$$

with stability parameters larger than κ , if the conditions

$$\xi^{\mu, \text{out}} h(I^\mu) > \kappa, \quad \text{with} \quad h(I^\mu) = \frac{1}{\sqrt{N}} \sum_j J_j \xi_j^\mu,$$

are satisfied simultaneously for all μ s. (Note the normalization factor $1/\sqrt{N}$. In this derivation the factor $1/N$ is shared between the field h and the interactions J_j .) All these conditions are expressed in the following equations:

$$\begin{cases} \prod_\mu \mathbf{1}\left(\frac{1}{\sqrt{N}} \xi^\mu \sum_j J_j \xi_j^\mu - \kappa\right) = 1 & \text{if } \{J_j\} \text{ is a solution,} \\ \prod_\mu \mathbf{1}\left(\frac{1}{\sqrt{N}} \xi^\mu \sum_j J_j \xi_j^\mu - \kappa\right) = 0 & \text{if } \{J_j\} \text{ is not a solution,} \end{cases}$$

where the suffix out is removed from $\xi^{\mu, \text{out}}$ for the sake of simplicity. $\mathbf{1}(x)$ is the step function:

$$\mathbf{1}(x) = \begin{cases} 1 & \text{if } x > 0, \\ 0 & \text{if } x \leq 0. \end{cases}$$

The volume Γ of the space of interactions is given by

$$\Gamma = \sum_{\{J_j\}} \prod_\mu \mathbf{1}\left(\frac{1}{\sqrt{N}} \xi^\mu \sum_j J_j \xi_j^\mu - \kappa\right),$$

where the summation runs over all possible values of the J_j s. The ranges of J_j s are restricted by the weak constraint

$$\sum_j (J_j)^2 = N.$$

This condition means that the norm of vector \bar{J} is fixed: $|\bar{J}|^2 = N$.

The normalization of vector \tilde{J} is necessary if one wants the volume of solutions to remain finite. With the normalization conditions the relative volume Γ becomes:

$$\Gamma = \frac{\int \prod dJ_j \left\{ \prod_{\mu} \left[1 \left(\frac{1}{\sqrt{N}} \xi^{\mu} \sum_j J_j \xi_j^{\mu} - \kappa \right) \right] \right\} \left[\delta \left(\sum_j (J_j)^2 - N \right) \right]}{\int \prod dJ_j \delta \left(\sum_j (J_j)^2 - N \right)}. \quad (6.22)$$

We note that Γ^T , the volume of solution for a complex neural network made of N visible units, is a product of N similar expressions, one for each neuron:

$$\Gamma^T = \prod_{i=1}^N \Gamma_i = (\Gamma)^N.$$

Because we are only interested in mean properties the expression (6.22) has to be averaged over all possible realizations of patterns I^{μ} . The calculations are carried out for random patterns which implies that the distribution of the components is given by

$$P(\xi_j^{\mu}) = \frac{1}{2} (\delta(\xi_j^{\mu} - 1) + \delta(\xi_j^{\mu} + 1)). \quad (6.23)$$

Sample averaging is an operation which must be performed on extensive quantities, otherwise the properties of a sample would depend on its size N according to complicated laws. One expects that relevant quantities are simple linear functions of N . This is why in statistical mechanics, according to Brout, sample averages have to be carried out on the free energy F of thermodynamical systems, which is indeed an extensive quantity, not on the partition function Z . That is to say, one must compute $\langle \log(Z) \rangle$ ($= -\beta \langle F \rangle$) and not $\langle Z \rangle$.

The same remark applies to the computation of the typical volume Γ^T : $\log(\Gamma^T)$ (not Γ^T) is the extensive quantity, since

$$\log(\Gamma^T) = N \log(\Gamma).$$

The quantity to be computed is therefore $\overline{\log(\Gamma)}$, and the typical volume is given by

$$\exp(\overline{\log(\Gamma)}).$$

One observes that we arrived at a problem which is symmetrical with respect to the spin glass problem, so to speak. In the spin glass problem a random set of interactions $\{J_{ij}\}$ is given and one looks for order parameters. Order parameters are combinations of dynamical variables σ_i which do not vanish in the limit of large N s and which minimize the free energy F . Here a set of spin states $\{\xi_j^{\mu}\}$, those defined by the patterns I^{μ} to be memorized, are given and one looks for macroscopic quantities, combinations of the dynamic variables J_j which minimize the ‘free energy’ which is computed starting from Eq. (6.22). The properties of spin glasses are embedded in two order parameters, the average magnetization M and the Edwards Anderson parameter Q :

$$M = \frac{1}{\sqrt{N}} \sum_j \langle \sigma_j \rangle \quad \text{and} \quad Q = \frac{1}{N} \sum_j \langle \sigma_j \rangle^2.$$

The physical meaning of the E.A. parameter is associated with the overlap degree between the various solutions of spin states. The same parameters are defined in the memorization problem with now the interaction J_j as the dynamic variable:

$$M = \frac{1}{\sqrt{N}} \sum_j \bar{J}_j \quad \text{and} \quad Q = \frac{1}{N} \sum_j (\bar{J}_j)^2.$$

Here Q is associated with the overlap between the various solutions for the interactions J_j : $Q = 1$ means that there is only one solution left ($\Gamma \rightarrow 0$). The space of possible solutions is then reduced to a single point and therefore $Q = 1$ is the sign that the maximum capacity of the network has been reached.

The calculation of the typical volume proceeds along lines which are similar to the ones followed by Amit *et al.* in their derivation of the memory storage capacity of Hebbian neural networks (see section 4.4).

a) Elimination of disorder

This is carried out by appealing to the replica trick; that is, by using the identity

$$\overline{\log(\Gamma)} = \lim_{n \rightarrow 0} \frac{\overline{\Gamma^n} - 1}{n}$$

where $\overline{\Gamma^n} = \mathcal{N}/\mathcal{D}$ and

$$\begin{aligned} \mathcal{N} &= \int \prod_{\alpha,j} dJ_j^\alpha \left[\left(\prod_\mu \mathbf{1} \left(\frac{1}{\sqrt{N}} \xi^\mu \sum_j J_j^\alpha \xi_j^\mu - \kappa \right) \right) \left[\delta \left(\sum_j (J_j^\alpha)^2 - N \right) \right] \right], \\ \mathcal{D} &= \int \prod_{\alpha,j} dJ_j^\alpha \delta \left(\sum_j (J_j^\alpha)^2 - N \right). \end{aligned} \quad (6.24)$$

The index α , $\alpha = 1, 2, \dots, n$ labels the replicas. We then introduce an integral representation of the step function:

$$\begin{aligned} \mathbf{1}(a - \kappa) &= \int_0^\infty d\lambda' \delta(\lambda' - (a - \kappa)) = \int_\kappa^\infty d\lambda \delta(\lambda - a) \\ &= \int_\kappa^\infty \frac{d\lambda}{2\pi} \int_{-\infty}^{+\infty} dx \exp(i x(\lambda - a)). \end{aligned}$$

With $a = \frac{1}{\sqrt{N}} \xi^\mu \sum_j J_j \xi_j^\mu$, the Eqs (6.24) become:

$$\begin{aligned} \mathcal{N} &= \int \prod_{j,\alpha} dJ_j^\alpha \prod_{\mu=1}^P \int_\kappa^\infty \frac{d\lambda_\mu^\alpha}{2\pi} \int_{-\infty}^{+\infty} dx_\mu^\alpha \\ &\quad \times \exp \left[ix_\mu^\alpha \left(\lambda_\mu^\alpha - \frac{\xi^\mu}{\sqrt{N}} \sum_j J_j^\alpha \xi_j^\mu \right) \right] \delta \left[\sum_j (J_j^\alpha)^2 - N \right], \\ \mathcal{D} &= \int \prod_{j,\alpha} dJ_j^\alpha \delta \left(\sum_j (J_j^\alpha)^2 - N \right). \end{aligned}$$

The sample average $\overline{(\dots)}$ is carried out in the numerator of Eq. (6.4) by using a random distribution of ξ_j^μ (Eq. 6.21). This yields

$$\begin{aligned} & \int d\xi_j^\mu P(\xi_j^\mu) \prod_\alpha \exp \left[-ix_\mu^\alpha \frac{\xi_j^\mu}{\sqrt{N}} \sum_j J_j^\alpha \xi_j^\mu \right] \\ &= \int d\xi_j^\mu P(\xi_j^\mu) \prod_j \exp \left[-\frac{i}{\sqrt{N}} \xi_j^\mu \sum_\alpha x_\mu^\alpha J_j^\alpha \right] \\ &= \prod_j \cos \left(\frac{1}{\sqrt{N}} \sum_\alpha x_\mu^\alpha J_j^\alpha \right) = \exp \sum_j \log \left[\cos \left(\frac{1}{\sqrt{N}} \sum_\alpha x_\mu^\alpha J_j^\alpha \right) \right], \end{aligned}$$

and, with the expansion $\log(\cos(x)) \simeq -\frac{1}{2}x^2$, one finds

$$\begin{aligned} & \int d\xi_j^\mu P(\xi_j^\mu) \prod_\alpha \exp \left[-ix_\mu^\alpha \frac{\xi_j^\mu}{\sqrt{N}} \sum_j J_j^\alpha \xi_j^\mu \right] \\ & \simeq \exp \left[-\frac{1}{2N} \sum_\alpha \sum_\beta x_\mu^\alpha x_\mu^\beta \sum_j J_j^\alpha J_j^\beta \right] \\ &= \exp \left[-\sum_\beta \sum_{\alpha<\beta} x_\mu^\alpha x_\mu^\beta \frac{1}{N} \sum_j J_j^\alpha J_j^\beta - \frac{1}{2} \sum_\alpha (x_\mu^\alpha)^2 \right]. \end{aligned}$$

The normalization $\sum_j (J_j^\alpha)^2 = N$ of efficacies has been used. Finally, the expression (6.24) implies

$$\begin{aligned} \overline{\Gamma^n} &= \left[\int \prod_{j,\alpha} dJ_j^\alpha \delta \left(\sum_j (J_j^\alpha)^2 - N \right) \right]^{-1} \\ & \quad \int \prod_{j,\alpha} dJ_j^\alpha \prod_{\mu=1}^P \int_\kappa^{+\infty} \frac{d\lambda_\mu}{2\pi} \int_{-\infty}^{+\infty} dx_\mu^\alpha \\ & \quad \exp \left\{ i \sum_\alpha x_\mu^\alpha \lambda_\mu^\alpha - \frac{1}{2} \sum_\alpha (x_\mu^\alpha)^2 \right. \\ & \quad \left. - \sum_\beta \sum_{\alpha<\beta} x_\mu^\alpha x_\mu^\beta \frac{1}{N} \sum_j J_j^\alpha J_j^\beta \right\} \delta \left[\sum_j (J_j^\alpha)^2 - N \right]. \end{aligned}$$

b) Introduction of order parameters

The integration over the variables J_j^α is split into two parts, an integration which keeps the order parameters $Q_{\alpha\beta}$, defined by

$$Q_{\alpha\beta} = \frac{1}{N} \sum_j J_j^\alpha J_j^\beta,$$

constant and an integration over these order parameters. The order parameters are defined for $\alpha < \beta$ strictly, which means in particular that their number is $\frac{1}{2}n(n-1)$.

The equation giving the typical volume becomes

$$\begin{aligned}\overline{\Gamma^n} = & \left[\int \prod_{j,\alpha} dJ_j^\alpha \delta \left(\sum_j (J_j^\alpha)^2 - N \right) \right]^{-1} \\ & \times \int \prod_{j,\alpha} dJ_j^\alpha \prod_\mu \int_\kappa^\infty \frac{d\lambda_\mu^\alpha}{2\pi} \int_{-\infty}^{+\infty} dx_\mu^\alpha \int_{-\infty}^{+\infty} \prod_\beta \prod_{\alpha<\beta} dQ_{\alpha\beta} \\ & \times \exp \left\{ i \sum_\alpha x_\mu^\alpha \lambda_\mu^\alpha - \frac{1}{2} \sum_\alpha (x_\mu^\alpha)^2 - \sum_\beta \sum_{\alpha<\beta} x_\mu^\alpha x_\mu^\beta Q_{\alpha\beta} \right\} \\ & \times \prod_\alpha \delta \left[\sum_j (J_j^\alpha)^2 - N \right] \prod_\beta \prod_{\alpha<\beta} \delta \left[\sum_j J_j^\alpha J_j^\beta - N Q_{\alpha\beta} \right].\end{aligned}$$

Using the integral representation of the delta function, the expression

$$\prod_\alpha \delta \left[\sum_j (J_j^\alpha)^2 - N \right] \times \prod_\beta \prod_{\alpha<\beta} \delta \left[\sum_j J_j^\alpha J_j^\beta - N Q_{\alpha\beta} \right]$$

can be written as

$$\begin{aligned}& \int_{-\infty}^{+\infty} \prod_\alpha \frac{dE_\alpha}{2\pi} \int_{-\infty}^{+\infty} \prod_\beta \prod_{\alpha<\beta} \frac{dF_{\alpha\beta}}{2\pi} \\ & \quad \times \exp i \left[E_\alpha \left(\sum_j (J_j^\alpha)^2 - N \right) + F_{\alpha\beta} \left(\sum_j J_j^\alpha J_j^\beta - N Q_{\alpha\beta} \right) \right] \\ &= \int_{-\infty}^{+\infty} \prod_\alpha \frac{dE_\alpha}{2\pi} \int_{-\infty}^{+\infty} \prod_\beta \prod_{\alpha<\beta} \frac{dF_{\alpha\beta}}{2\pi} \\ & \quad \times \exp \left\{ iN \left[\sum_\alpha (E_\alpha (J^\alpha)^2 - E_\alpha) + \sum_\beta \sum_{\alpha<\beta} (F_{\alpha\beta} J_\alpha J_\beta - F_{\alpha\beta} Q_{\alpha\beta}) \right] \right\}.\end{aligned}$$

In this expression the index j has been skipped because the integral over the sites factorizes. The parameters E_α and $F_{\alpha\beta}$ play the role of Lagrange multipliers, as explained in section 4.2.4. As the integral over the patterns μ also factorizes, one can write, using expressions such as

$$\begin{aligned}\prod_\alpha f_\alpha &\equiv \exp \left[\log \left(\prod_\alpha f_\alpha \right) \right], \\ \overline{\Gamma^n} &= \left[\int_{-\infty}^{+\infty} \prod_\alpha \frac{dE_\alpha}{2\pi} \exp(N G^0) \right]^{-1} \int_{-\infty}^{+\infty} \prod_\alpha \frac{dE_\alpha}{2\pi} \\ & \quad \times \int_{-\infty}^{+\infty} \prod_\beta \prod_{\alpha<\beta} \frac{dF_{\alpha\beta}}{2\pi} \int_{-\infty}^{+\infty} \prod_\beta \prod_{\alpha<\beta} \frac{dQ_{\alpha\beta}}{2\pi} \exp(NG),\end{aligned}$$

with $\alpha = P/N$ and

$$\begin{aligned}G(Q_{\alpha\beta}, F_{\alpha\beta}, E_\alpha) &= \alpha G_1(Q_{\alpha\beta}) + G_2(F_{\alpha\beta}, E_\alpha) \\ & \quad - i \sum_\beta \sum_{\alpha<\beta} F_{\alpha\beta} Q_{\alpha\beta} - i \sum_\alpha E_\alpha,\end{aligned}$$

$$G^0 = G_2(0, E_\alpha) - i \sum_\alpha E_\alpha. \tag{6.25}$$

G_1 and G_2 are given by

$$\begin{aligned} G(Q_{\alpha\beta}) &= \dots \\ &= \log \left\{ \int_{-\infty}^{+\infty} \prod_{\alpha=1}^n dx^\alpha \int_{\kappa}^{\infty} \prod_{\alpha=1}^n \frac{d\lambda^\alpha}{2\pi} \right. \\ &\quad \times \exp \left[i \sum_{\alpha} x^\alpha \lambda^\alpha - \frac{1}{2} \sum_{\alpha} (x^\alpha)^2 - \sum_{\alpha<\beta} x^\alpha x^\beta Q_{\alpha\beta} \right] \Bigg\}, \\ G_2(F_{\alpha\beta}, E_\alpha) &= \log \left\{ \prod_{\alpha=1}^n \int_{-\infty}^{+\infty} dJ_\alpha \exp i \left[\sum_{\alpha} E_\alpha (J_\alpha)^2 + \sum_{\alpha<\beta} F_{\alpha\beta} J_\alpha J_\beta \right] \right\}. \end{aligned}$$

c) *Saddle point equations and replica symmetry approximation*

According to the discussion on the saddle point method in section 4.2.4, the integrals are dominated by the saddle point of the exponent with respect to the Lagrange parameters E_α , $F_{\alpha\beta}$ and $Q_{\alpha\beta}$. In other words, one has to solve the following equations:

$$\frac{\partial G}{\partial E_\alpha} = 0, \quad \frac{\partial G}{\partial F_{\alpha\beta}} = 0, \quad \frac{\partial G}{\partial Q_{\alpha\beta}} = 0 \quad \forall \alpha, \beta.$$

G is defined in Eq. (6.25). The saddle point equation $\partial G^0 / \partial E_\alpha = 0$ only yields an irrelevant constant.

The equilibrium equations are reduced to three equations by the replica symmetry ansatz

$$Q_{\alpha\beta} = Q, \quad F_{\alpha\beta} = F, \quad E_\alpha = E, \quad \forall \alpha, \beta, \quad (\beta < \alpha),$$

and all that remains to be computed are the two integrals I_1 and I_2 :

$$\begin{aligned} I_1 &= \int_{-\infty}^{+\infty} \prod_{\alpha} dx_\alpha \int_{\kappa}^{\infty} \prod_{\alpha} \frac{d\lambda^\alpha}{2\pi} \exp \left[i \sum_{\alpha} x_\alpha \lambda_\alpha - \frac{1}{2} \sum_{\alpha} x_\alpha^2 - Q \sum_{\beta} \sum_{\alpha<\beta} x_\alpha x_\beta \right], \\ I_2 &= \int_{-\infty}^{+\infty} \prod_{\alpha} dJ_\alpha \exp \left\{ i \left[E \sum_{\alpha} J_\alpha^2 + F \sum_{\beta} \sum_{\alpha<\beta} J_\alpha J_\beta \right] \right\}. \end{aligned}$$

- Using the identity $\sum_{\beta} \sum_{\alpha<\beta} x_\alpha x_\beta \equiv \frac{1}{2} [(\sum_{\alpha} x_\alpha)^2 - \sum_{\alpha} (x_\alpha)^2]$, the first integral becomes

$$\begin{aligned} I_1 &= \int_{-\infty}^{+\infty} \prod_{\alpha} dx_\alpha \int_{\kappa}^{\infty} \prod_{\alpha} \frac{d\lambda^\alpha}{2\pi} \\ &\quad \times \exp \left\{ \sum_{\alpha} (ix_\alpha \lambda_\alpha - \frac{1}{2} (x_\alpha)^2 (1 - Q)) - \frac{1}{2} Q \left[\sum_{\alpha} x_\alpha \right]^2 \right\}. \end{aligned}$$

We apply the Gauss transform to the last term of the exponent:

$$\exp \left(-\frac{1}{2} ax^2 \right) \equiv \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} dy \exp \left(-\frac{1}{2} y^2 + i\sqrt{a} xy \right).$$

I_1 is then decomposed into n identical decoupled integrals:

$$G_1(Q) = \log \int_{-\infty}^{+\infty} \frac{dy}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}y^2\right) H^n(Q),$$

with $H = \int_{-\infty}^{+\infty} dx \int_{\kappa}^{+\infty} \frac{d\lambda}{2\pi} \exp\left(-\frac{1}{2}x^2(1-Q) + ix(\lambda + y\sqrt{Q})\right).$

For $n \rightarrow 0$, $G_1(Q)$ becomes

$$\begin{aligned} G_1(Q) &= \log \int_{-\infty}^{+\infty} \frac{dy}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}y^2\right) \exp(n \log H) \\ &\simeq \log \int_{-\infty}^{+\infty} \frac{dy}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}y^2\right) (1 + n \log H) \\ &= \log \left[1 + n \int_{-\infty}^{+\infty} \frac{dy}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}y^2\right) \log H \right], \end{aligned}$$

whence

$$\begin{aligned} G_1(Q) &= n \int_{-\infty}^{+\infty} \frac{dy}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}y^2\right) \\ &\quad \times \log \left\{ \int_{-\infty}^{+\infty} dx \int_{\kappa}^{+\infty} \frac{d\lambda}{2\pi} \exp\left[-\frac{1}{2}x^2(1-Q) + ix(\lambda + y\sqrt{Q})\right] \right\}. \end{aligned}$$

The exponent is transformed:

$$\begin{aligned} &\exp\left[-\frac{1}{2}(1-Q)\left(x^2 - 2ix\frac{(\lambda + y\sqrt{Q})}{1-Q}\right)\right] \\ &= \exp\left\{-\frac{1}{2}(1-Q)\left[x - i\left(\frac{\lambda + y\sqrt{Q}}{1-Q}\right)\right]^2\right\} \exp\left[-\frac{1}{2}\frac{1}{1-Q}(\lambda + y\sqrt{Q})^2\right]; \end{aligned}$$

and the summation over the variable x is carried out:

$$\begin{aligned} \int_{-\infty}^{+\infty} dx \exp\left[-\frac{1}{2}(1-Q)(x - ix_0)^2\right] &= \int_{-\infty - ix_0}^{+\infty - ix_0} dt \exp\left[-\frac{1}{2}(1-Q)t^2\right] \\ &= \int_{-\infty}^{+\infty} dt \exp\left[-\frac{1}{2}(1-Q)t^2\right] = \sqrt{2\pi} \sqrt{1-Q}. \end{aligned}$$

One obtains

$$\begin{aligned} G_1(Q) &= n \int_{-\infty}^{+\infty} \frac{dy}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}y^2\right) \\ &\quad \times \log \left\{ \int_{\kappa}^{+\infty} \frac{d\lambda}{\sqrt{2\pi}} \frac{1}{\sqrt{1-Q}} \exp\left[-\frac{1}{2}\frac{1}{1-Q}(\lambda + y\sqrt{Q})^2\right] \right\}, \end{aligned}$$

which, with the following change of variable $z = \frac{\lambda + y\sqrt{Q}}{\sqrt{1-Q}}$, becomes

$$G_1(Q) = n \int_{-\infty}^{+\infty} \frac{dy}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}y^2\right) \log \int_{(\kappa+y\sqrt{Q})/\sqrt{1-Q}}^{+\infty} dz \exp\left(-\frac{1}{2}z^2\right),$$

an expression which is rewritten as

$$G_1(Q) = n \int D_y \log \left[H\left(\frac{\kappa + y\sqrt{Q}}{\sqrt{1-Q}}\right) \right], \quad (6.26)$$

with $H(x) = \int_x^{+\infty} Dz \quad \text{and} \quad Dz = \frac{dz}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}z^2\right).$

- The second integral I_2 may also be obtained by appealing to the same technique, namely the Gaussian transform, that we used in the calculation of I_1 . We find it more instructive to introduce another approach in which we proceed by a direct diagonalization of the exponent of I_2 . One has

$$\exp\left\{ i \left[E \sum_{\alpha} (J_{\alpha})^2 + F \sum_{\beta} \sum_{\alpha < \beta} J_{\alpha} J_{\beta} \right] \right\} = \exp(i \bar{J} \cdot \mathbf{X} \cdot \bar{J}),$$

where the $n \times n$ matrix \mathbf{X} is given by

$$\mathbf{X} = \begin{pmatrix} E & \frac{1}{2}F & \frac{1}{2}F & \cdots & \frac{1}{2}F \\ \frac{1}{2}F & E & \frac{1}{2}F & & \\ \frac{1}{2}F & \frac{1}{2}F & E & & \\ \vdots & & & \ddots & \\ \frac{1}{2}F & & & & E \end{pmatrix}$$

The matrix has

$$\begin{cases} \text{a non-degenerate eigenvalue } \lambda_0 = E + \frac{1}{2}(n-1)F, \\ \text{an } (n-1)\text{-degenerate eigenvalue } \lambda_1 = E - \frac{1}{2}F. \end{cases}$$

In the system of eigen-axes of \mathbf{X} the calculation of I_2 reduces to that of a product of Gaussian integrals. One has

$$\begin{aligned} \int_{-\infty}^{+\infty} dx \exp(i\lambda x^2) &= e^{i\pi/4} \int_{-e^{i\pi/4}\infty}^{+e^{i\pi/4}\infty} dx \exp(-\lambda x^2) \\ &= e^{i\pi/4} \int_{-\infty}^{+\infty} dx \exp(-\lambda x^2) = e^{i\pi/4} \left(\frac{\pi}{\lambda}\right)^{1/2} = \left(\frac{i\pi}{\lambda}\right)^{1/2}, \end{aligned}$$

whence

$$I_2 = (i\pi)^{n/2} \frac{1}{\prod_{\alpha=1}^n \lambda_{\alpha}^{1/2}} = (i\pi)^{n/2} \frac{1}{\lambda_0^{1/2} \lambda_1^{(n-1)/2}}.$$

This can be written as

$$\begin{aligned} (\mathrm{i}\pi)^{-n/2} I_2 &= \exp -\frac{1}{2} [\log(E + \frac{1}{2}(n-1)F) + (n-1)\log(E - \frac{1}{2}F)] \\ &= \exp -\frac{1}{2} [n \log(E - \frac{1}{2}F) + \log(1 + n \frac{\frac{1}{2}F}{E - \frac{1}{2}F})] \\ &\simeq \exp \frac{1}{2}n \left[-\log(E - \frac{1}{2}F) - \frac{\frac{1}{2}F}{E - \frac{1}{2}F} \right], \end{aligned}$$

and therefore, skipping irrelevant constants,

$$G_2(F, E) = \frac{1}{2}n \left[-\log(E - \frac{1}{2}F) - \frac{\frac{1}{2}F}{E - \frac{1}{2}F} \right]. \quad (6.27)$$

Finally, G is given by

$$\begin{aligned} G &= \alpha n \int Dy \log \left[H \left(\frac{\kappa + y\sqrt{Q}}{\sqrt{1-Q}} \right) \right] \\ &\quad - \frac{1}{2}n \left[\log(E - \frac{1}{2}F) + \frac{\frac{1}{2}F}{E - \frac{1}{2}F} \right] - n\mathrm{i}E - \frac{1}{2}n(n-1)\mathrm{i}FQ. \end{aligned} \quad (6.28)$$

In the limit $n \rightarrow 0$ the last term is $\frac{1}{2}n\mathrm{i}FQ$.

One gets rid of the variables E and F by using the saddle point equations,

$$\frac{\partial}{\partial E} (G_2(F, E) - n\mathrm{i}E + \frac{1}{2}n\mathrm{i}FQ) = 0, \quad \text{whence} \quad \frac{\frac{1}{2}F}{(E - \frac{1}{2}F)^2} - \frac{1}{E - \frac{1}{2}F} - 2\mathrm{i} = 0,$$

$$\frac{\partial}{\partial F} (G_2(F, E) - n\mathrm{i}E + \frac{1}{2}n\mathrm{i}FQ) = 0, \quad \text{whence} \quad \frac{\frac{1}{2}F}{(E - \frac{1}{2}F)^2} - 2\mathrm{i}Q = 0.$$

Substracting the two equations, one finds

$$E - \frac{1}{2}F = \frac{1}{2}\mathrm{i} \frac{1}{(1-Q)} \quad \text{and} \quad \frac{\frac{1}{2}F}{E - \frac{1}{2}F} = -\frac{-Q}{(1-Q)}.$$

As for the last two terms in G , one arrives at

$$\begin{aligned} -\mathrm{i}E + \frac{1}{2}FQ &= -\mathrm{i} \left(E - \frac{1}{2}F + \frac{1}{2}F(1-Q) \right) \\ &= -\mathrm{i} \left(E - \frac{1}{2}F \right) \left(1 + (1-Q) \left(\frac{-Q}{1-Q} \right) \right) = 1. \end{aligned}$$

Skipping constants which cancel out in the derivative of G with respect to Q , one finally finds

$$G(Q) = \alpha \times \int Dy \log \left[H \left(\frac{\kappa + y\sqrt{Q}}{\sqrt{1-Q}} \right) \right] + \frac{1}{2} \log(1-Q) + \frac{1}{2} \frac{Q}{1-Q}. \quad (6.29)$$

- The solution is given by the last saddle point condition

$$\frac{dG(Q)}{dQ} = 0, \quad \text{whence} \quad \frac{Q}{2(1-Q)^2} = -\alpha \times \int Dy \frac{d}{dQ} \left\{ \log \left[H \left(\frac{\kappa + y\sqrt{Q}}{\sqrt{1-Q}} \right) \right] \right\}.$$

It is interesting to study the behavior of this equation in the limit $Q \rightarrow 1$ since we have argued at the beginning of this section that the network reaches its limit memory storage capacity when $Q = 1$. In this limit the argument of H goes to $+\infty$ if $y > -\kappa$. H is then close to zero and $\log(H)$ diverges. If $y < -\kappa$ the argument of H goes to $-\infty$. H is of the order of 1 and $\log(H)$ vanishes. The integral in Eq. (6.29) is dominated by values $y > -\kappa$.

Letting $x = \frac{\kappa + y\sqrt{Q}}{\sqrt{1-Q}}$, one has

$$\frac{Q}{2(1-Q)^2} = \alpha \int_{-\kappa}^{+\infty} Dy \frac{H(x)}{H'(x)} \frac{\partial x}{\partial Q},$$

with

$$H(x) = \int_x^{+\infty} \frac{dz}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}z^2\right) \quad \text{and} \quad H'(x) = -\frac{\exp\left(-\frac{1}{2}x^2\right)}{\sqrt{2\pi}}.$$

For $y > -\kappa$, $x \rightarrow \infty$ when $Q \rightarrow 1$. In this limit,

$$H(x) = \frac{1}{2} \left[1 - \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right) \right] \simeq \frac{1}{\sqrt{2\pi}} \frac{1}{x} \exp\left(-\frac{1}{2}x^2\right),$$

and therefore

$$\frac{Q}{2(1-Q)^2} \simeq \alpha \int_{-\kappa}^{+\infty} \frac{dy}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}y^2\right) \left[x(Q, y) \frac{\partial x(Q, y)}{\partial Q} \right].$$

With $\frac{\partial x}{\partial Q} = \frac{1}{2(1-Q)} [x + y\sqrt{(1-Q)/Q}]$, this equation can be written as

$$Q = \alpha \int_{-\kappa}^{+\infty} \frac{dy}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}y^2\right) [\kappa + y\sqrt{Q}] \left[\kappa + \frac{y}{\sqrt{Q}} \right].$$

It has to be taken in the limit $Q \rightarrow 1$. Then it yields the storage capacity α_c of the network,

$$1 = \alpha_c(\kappa) \int_{-\kappa}^{+\infty} \frac{dy}{2\pi} \exp\left(-\frac{1}{2}y^2\right) [\kappa + y]^2,$$

which is Eq. (6.21).

LEARNING DYNAMICS IN 'VISIBLE' NEURAL NETWORKS

Computing the volume of solutions $\Gamma(\mathcal{P})$ of a given problem \mathcal{P} , an art which is still in its infancy, yields invaluable information on the possibility of using a neural network to solve \mathcal{P} . But it is not enough to know that a solution $\tilde{\mathbf{J}}$ exists. A solution must positively be found, which amounts to saying that one has to devise learning dynamics which bring $\tilde{\mathbf{J}}$ inside Γ . It must be noted that not all acceptable learning dynamics are equally efficient. For example, the Hebbian learning rule is good at giving a solution to the memorization of P random patterns in an N -neural network, at least when P is less than $0.14 \times N$. It fails for larger values of P even if solutions still exist up to $P = 2 \times N$. Hebbian rules may be improved, however, and it is possible to find learning dynamics which allow the limit capacity to be reached.

On the other hand one can be less concerned with the limitation of memory storage capacity than with the phenomenon of overcrowding catastrophe, which occurs when the limit capacity is trespassed.

Solutions for both problems, that of improving memory storage capacities and that of avoiding the overcrowding catastrophe, are put forward in this chapter.

7.1 A classification of learning dynamics

Generally speaking, learning is defined as the ability for a neural network to change its parameters, synaptic efficacies or threshold levels so as to improve its response with respect to its environment. This is the subtle interplay between the dynamics of states and the dynamics of parameters which makes the study of neural networks so attractive and also so difficult. By environment we mean any structure which is external to the neural network and which may influence its state. Environment therefore may be internal to the creature which houses the neural network as well: any organism has a number of vital needs which must be absolutely satisfied. When such a need appears, a specialized structure of the organism sends a series of messages to the nervous system which must react in such a way as to reduce the inconvenience. The environment

may be the natural surroundings of the animal too. In that case its influence on the neural network is passive and the organism has to learn the rules which structure its ecological niche. Finally, the environment may be active. It takes the form of a teacher who can change the material to be learnt according to the performances of the learning organism.

Two main classes of learning, namely *unsupervised learning* and *supervised learning*, are generally distinguished (Fig. 7.1).

a) *Unsupervised learning*. — It must be reminded that neural networks comprise three types of units, the input and output units, making up the visible units, and the hidden units. In unsupervised learning the set \mathcal{E} of patterns I^μ to be memorized is a set of *input patterns*:

$$I^\mu \equiv I^{\mu, \text{in}} = \{\xi_{i \in \mathcal{I}}^\mu\}, \quad \xi_i^\mu \in \{-1, +1\}.$$

The system must therefore solve the problem of credit assignment, that is to say the problem of determining by itself the states of its hidden units and output units (which may be considered here as a subset of hidden units). This process is called *self-organization*. The way the system is modified during the learning stage depends on the learning algorithm, on the topological constraints that the network has to satisfy and on the learning set. The first two features are under genetic control and successful learning is the result of natural selection. Self-organization is of paramount importance during the epigenesis of the nervous systems. This type of learning is studied in Chapter 9.

b) *Supervised learning*. — In supervised learning \mathcal{E} is a set of *visible patterns*

$$I^\mu = I^{\mu, \text{in}} \otimes I^{\mu, \text{out}},$$

which means that one knows what the output states should be. The training is supervised in that all states of visible units always remain under the control of the learning process. There exist two sorts of supervised learning:

- Learning may be the result of an *association process*. During the learning session the visible states (the input states and the ouput states) are fully determined by the patterns of the learning set \mathcal{E} . The synaptic efficacies are automatically modified according to some variant of the Hebbian association mechanism. The performance of learning is checked only after the training session is over. There is no control on the performance of learning during the training session itself.

- On the other hand learning may be considered as the result of an *error correction process*. The response $\sigma_{i \in \mathcal{O}}(I^\mu)$ of the system to a given stimulus $I^{\mu, \text{in}}$ is compared with what is desired, that is $I^{\mu, \text{out}}$. When

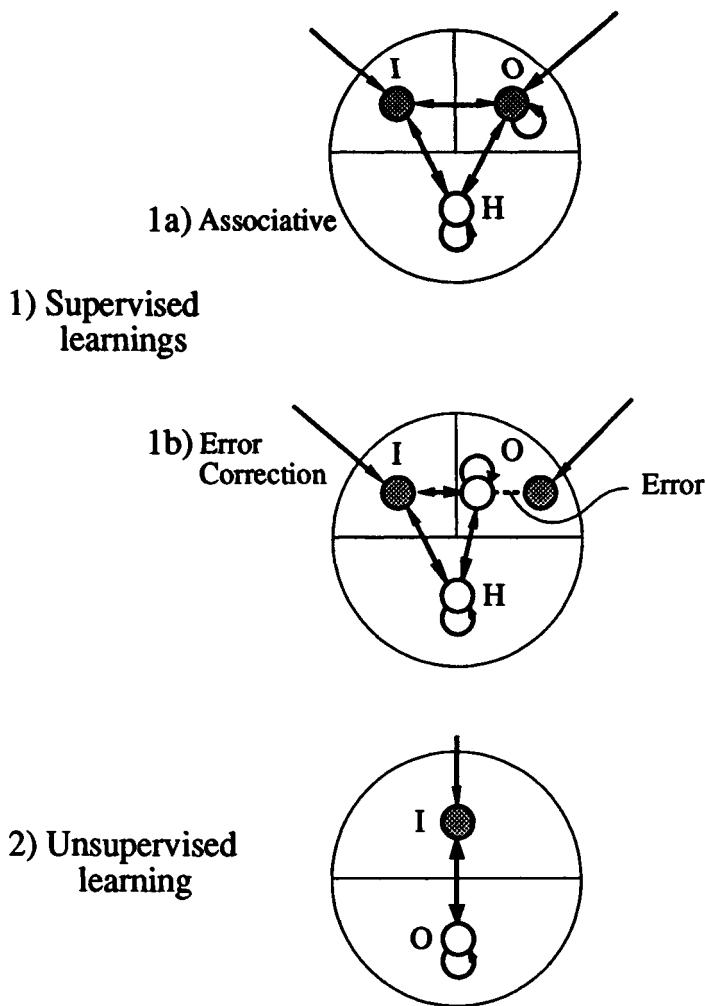


Figure 7.1. Learning paradigms (shaded circles are for fixed states).

1.a) Associative learning: input and output states are clamped.

1.b) Error correction: observed and desired outputs are compared.

2) Unsupervised learning: the system self-organizes.

the answer to a given stimulus $I^{\mu, \text{in}}$ is non-satisfactory, the synaptic efficacies are modified so as to make the right answer to that stimulus more likely. In some cases one is able to relate analytically the values of these modifications to those of signal errors. One is then certain that the modifications bring the system in the right direction. However, it

is not always possible to foresee the outcome of a given variation of the parameters of the system and therefore to decide *a priori* how they must be changed. The way out is to proceed through *trial and error, reward and punishment*. For example, the system undergoes a slight random modification of its synaptic efficacies. The change is kept if the answer to a given stimulus is improved. It is given up if the answer has deteriorated. The various types of learning are summarized below:

$$\text{Learning is } \begin{cases} \text{unsupervised} & (\text{self-organization}), \\ \text{supervised} & \begin{cases} \text{association paradigm}, \\ \text{error correction paradigm}. \end{cases} \end{cases}$$

7.2 Constraining the synaptic efficacies

In the rest of this chapter we assume that the *system is made of visible units exclusively*. The network experiences, one by one and at regular intervals of time, the patterns of a series of patterns to be memorized I^μ , $\mu = 1, 2, \dots, P$. Let $J_{ij}(\mu - 1)$ be the synaptic efficacies after the pattern $I^{\mu-1}$ has been learned. The symmetrical Hebbian learning dynamics can be written as

$$J_{ij}(\mu) = J_{ij}(\mu - 1) + \Delta J_{ij},$$

$$\text{with } \Delta J_{ij} = \varepsilon \xi_i^\mu \xi_j^\mu, \quad \xi_i^\mu \in \{+1, -1\}, \quad \varepsilon > 0. \quad (7.1)$$

Once the system has experienced the whole string of P patterns, its synaptic efficacies are given by

$$J_{ij} = \varepsilon \sum_{\mu=1}^P \xi_i^\mu \xi_j^\mu.$$

This set of interactions defines the Hopfield model which has been thoroughly studied in Chapter 4. If the set \mathcal{E} of patterns is learned again and again, the synaptic efficacies grow beyond any limit. However, the range of efficacies of real synapses is necessarily restricted. Several mechanisms have been proposed; some involve limited cellular resources, these being weak constraints, while others account for limited synaptic resources, these being strong constraints. The biological nature of these resources is not known; it could be the number of postsynaptic neuroreceptor molecules or some other biochemical material such as the MAP2 proteins which are involved in the early plasticity of the visual system of cats.

7.2.1 Weak constraints

These constraints use some form of normalization for the set of efficacies of *synapses impinging on a given neuron i*. For example, Fukushima

(and also Von der Malsburg) use a Euclidean norm,

$$\sum_j |J_{ij}|^2 = L. \quad (7.2)$$

In statistical mechanics this condition gives rise to the so-called spherical models: for example L has been set to $L = N$ in the computation of Γ (in section 6.3). In actual fact weak constraints merely renormalize the synaptic efficacies and therefore they have no effects on the memorization properties of neural networks. This statement, which may seem obvious, is given a firmer basis below.

The learning dynamics of weakly constrained neural networks

From a computational point of view a learning step consists in imprinting the set \mathcal{E} of patterns and then renormalizing the synaptic efficacies so as to satisfy the constraint (7.2). The process may be written as

$$\Delta J_{ij} = J_{ij}^H - \frac{1}{\tau_i} J_{ij}, \quad (7.3)$$

where J_{ij}^H is the usual Hebbian rule $J_{ij}^H = \epsilon \sum_\mu \xi_i^\mu \xi_j^\mu$.

Equation (7.3) means that taking the weak constraint into account is equivalent to introducing a relaxation term $-J_{ij}/\tau_i$ (sometimes called a forgetting term) in the learning dynamics given by Eq. (7.1). To make this point conspicuous one computes $\sum_j J_{ij} \Delta J_{ij}$:

$$\begin{aligned} \sum_j J_{ij} \Delta J_{ij} &= \sum_j J_{ij} \left[J_{ij}^H - \frac{1}{\tau_i} J_{ij} \right] \\ &= \epsilon \sum_\mu \xi_i^\mu \sum_j J_{ij} \xi_j^\mu - \frac{1}{\tau_i} \sum_j (J_{ij})^2 \\ &= \epsilon \sum_\mu x_i^\mu - \frac{L}{\tau_i}. \end{aligned}$$

If one chooses $\tau_i^{-1} = \epsilon/L \sum_\mu x_i^\mu$, the expression vanishes,

$$\sum_j J_{ij} \Delta J_{ij} = 0,$$

and the norm of synaptic efficacies remains unchanged during learning. The fixed point of Eq. (7.3) is

$$J_{ij}^* = J_{ij}^H \tau_i = \frac{L J_{ij}^H}{\epsilon \sum_\mu x_i^\mu} = \frac{L J_{ij}^H}{\epsilon \sum_k J_{ik}^* \sum_\mu \xi_i^\mu \xi_k^\mu} = \frac{L J_{ij}^H}{\sum_k J_{ik}^* J_{ik}^H}.$$

To solve this equation in J_{ij}^* both sides of this equation are multiplied by J_{ij}^H and summed over j . This leads to

$$\left(\sum_j J_{ij}^* J_{ij}^H \right)^2 = L \sum_j (J_{ij}^H)^2$$

and therefore to $J_{ij}^* = k_i J_{ij}^H$, with $k_i = \sqrt{L}/\sqrt{\sum_j (J_{ij}^H)^2} > 0$.

Since the stability conditions

$$\sum_j J_{ij}^* \xi_j^\mu > 0$$

are not modified when all J_{ij} ’s of j synapses impinging on a given neuron i are transformed into $k_i J_{ij}$ with $k_i > 0$, the weak constraint (7.1) has no influence on the memory storage capacity of the network whatsoever.

Strict weak constraints defined by Eq. (7.2) can be replaced by loose weak constraints:

$$\sum_j J_{ij}^2 \leq L,$$

which allows the network to learn according to the pure Hebbian rule, (that is without any forgetting term), or to any other rule until the limits are reached.

7.2.2 Strong constraints

In these models the resources of *every synapse* are limited. The constraint may be loose or it may be strict.

- For strict strong constraints J_{ij} is modified according to a pure Hebbian rule as long as $|J_{ij}| < L$ and does not change any more when one of the two limits $+L$ or $-L$ is reached (see Fig. 7.2.b). Once all efficacies have arrived at saturation points the network interactions are frozen permanently. *Learning is irreversible*. Since these are the very first learned patterns which eventually may be memorized the rule provides a *model of primacy effects* (see section 2.4.4). The capacity critically depends on the ratio ε/L : for large ε ’s the limits are already obtained for the first pattern and therefore the capacity is $P = 1$. On the other hand, for very small ε ’s the overcrowding catastrophe occurs before the effect of limits become effective and $P = 0$. In between these two regimes there exists an optimal ratio ε_c/L allowing a maximum number of patterns to be memorized.

- When the constraints on boundaries are loose the learning rule still hinders any exploration of efficacies into the forbidden zones $|J_{ij}| > L$ (see Fig. 7.2.a) but the efficacies are never frozen and the rule is *reversible*. If ε is very large the synaptic efficacies are clipped at $+/-L$ at every learning step: the system memorizes only the very last pattern it experiences. Smaller values of ε allow the imprinting of the few last patterns of the learning set. We have a *model of recency effects*. If ε becomes too small the system is overcrowded before the synaptic efficacies arrive at their limits $+/-L$. The overcrowding catastrophe occurs as it does in the unconstrained model. Therefore there must exist an optimal ratio ε/L which makes the number of recently imprinted patterns as large as possible. The process by which the learning dynamics gradually replaces the formerly imprinted patterns by those which are learned last is

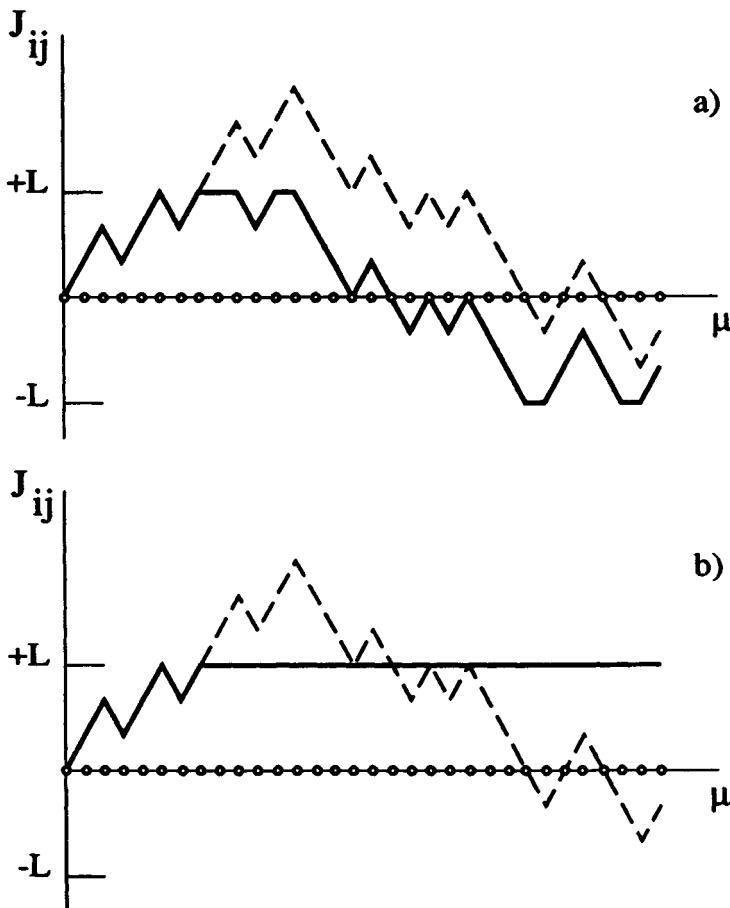


Figure 7.2. Two types of strong constraints for synaptic efficacies. a) Reversible mechanism. The efficacies are not allowed to overshoot the limits $+/-L$ but any move which brings them back into the available range is permitted. b) Irreversible mechanism. The efficacies stick to $+L$ or to $-L$ as soon as their values reach the limits.

called a palimpsest effect. This is a *model of short memory*. Palimpsests are parchments which monks in the Middle Ages used and re-used by scrubbing their surfaces. Erasing was never perfect and the parchments bore several less and less readable layers of handwriting.

One now gives some scaling arguments regarding how to choose an optimal limit L . Let $J_{ij}(\mu - 1)$ be the synaptic efficacies at a given moment of the learning session.

We want the system to learn a new pattern I^μ . Using the Hebbian rule, the efficacies become

$$J_{ij}(\mu) = J_{ij}(\mu - 1) + \varepsilon \xi_i^\mu \xi_j^\mu.$$

The local field is

$$h_i(I^\mu) = \sum_j J_{ij}(\mu) \xi_j^\mu = \varepsilon \xi_i^\mu \sum_j 1 + \sum_j J_{ij}(\mu - 1) \xi_j^\mu$$

and

$$x_i^\mu = h_i(I^\mu) \xi_i^\mu = \varepsilon N + \sum_j J_{ij}(\mu - 1) \xi_i^\mu \xi_j^\mu.$$

The pattern I^μ is well imprinted if the polarization parameter x_i^μ is positive. For this to be true it is necessary that

$$(\varepsilon N)^2 > N \langle \delta J^2 \rangle, \quad (7.4)$$

where $\langle \delta J^2 \rangle$ is the mean square fluctuation of the one synaptic efficacy. Therefore the critical parameter ε_c is given by

$$\varepsilon_c = \sqrt{\frac{\langle \delta J^2 \rangle}{N}}.$$

When the patterns are piled up into the network $\langle \delta J^2 \rangle$ increases and the critical parameter ε_c increases. The system becomes overcrowded when $\varepsilon_c > \varepsilon$ since the stability condition (7.4) is no more fulfilled. If $\sqrt{\langle \delta J^2 \rangle}$ is of the order of L before this occurs, the system does not get overcrowded. This is achieved for large enough ε 's. However, if ε is too large the memory is not optimally used. Optimality then requires that

$$\varepsilon = e \frac{L}{\sqrt{N}},$$

where e is some universal parameter. If one imposes $\varepsilon = 1/N$ (for the energy to be an extensive quantity) the limit value L must behave as

$$L = \frac{1}{e \sqrt{N}}.$$

In models of constrained efficacies the memory storage capacity is defined as the maximum number of patterns which are retained either by primacy or by recency effects. It is obtained when $\sqrt{\langle \delta J^2 \rangle}$ is of the order of L . As the synaptic weight fluctuations are given by

$$\langle \delta J \rangle^2 \simeq \frac{P}{N^2},$$

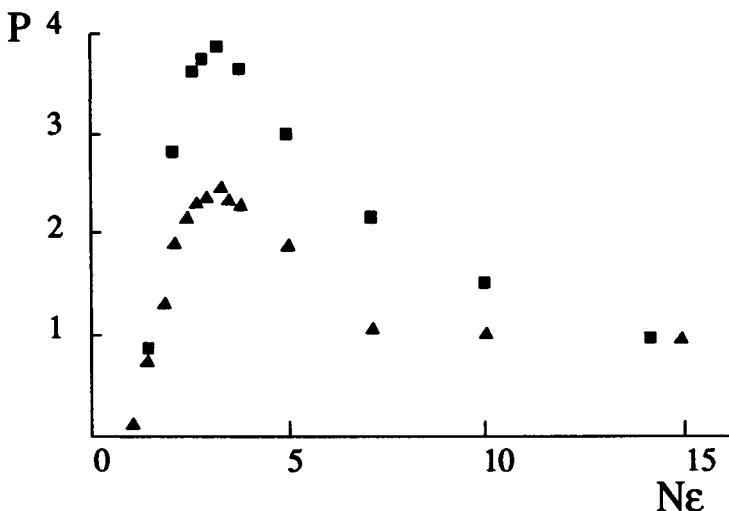


Figure 7.3. Computer simulations carried out on neural networks with $N = 100$ (triangles) and $N = 200$ (squares), showing the memory capacities of reversibly constrained systems (After Nadal *et al.*).

the limit capacity P_c is

$$\frac{P_c}{N^2} \simeq L^2 \simeq \frac{1}{N} \quad \text{or} \quad P_c = \alpha_c \times N.$$

The memory storage capacities are still proportional to the size N of the network but, as calculations and simulations show, the coefficient is appreciably reduced with respect to pure Hebbian models.

The model with reversible barriers has been investigated by Nadal, Dehaene, Changeux and Toulouse. Simulations show that $e \simeq 3.0$ and $\alpha_c \simeq 0.016$ (see Fig. 7.3). The ‘magic number’ of seven items, which the short-term memory should be able to store temporarily, would correspond to pools of neurons comprising about 500 neurons, which is of the order of the size of microcolumns. Moreover the experimental psychology experiments have shown that the erasing of this memory is more likely explained by a palimpsest mechanism than by a mere passive forgetting phenomenon.

The model with reversible limits and the model with irreversible limits have been solved analytically by M. Gordon, who emphasized an analogy between the synaptic evolution of such constrained neural networks and the properties of random walks confined by barriers. For reversible limits she recovers the value $e = 3.0$ observed in the simulations of Nadal. On the other hand she finds $\alpha_c = 0.045$, which is appreciably larger than the

value given by simulations. Simulations, however, have been carried out on very small systems (since P is of the order of 2 to 4), far from the range of validity of the theory. For irreversible limits Gordon finds $e = 3.33$ and $\alpha_c = 0.04$.

A closely related model, called the marginalist model, has been solved analytically by Nadal and Mézard, who used the replica technique we exposed in Chapter 4.

7.3 Projection algorithms

7.3.1 The Widrow Hoff algorithm

In the last section we considered some means of avoiding the overcrowding catastrophe. This goal was achieved at the expense of the memory storage capacities. In this section we consider the other problem, that of increasing the capacity. The Widrow Hoff approach is one of the most simplest ways of tackling this problem.

Let us consider once more the Hebbian synaptic dynamics. The modification brought about by the pattern I^μ transforms the synaptic efficacies according to

$$J_{ij}(\mu) = J_{ij}(\mu - 1) + \frac{1}{N} \xi_i^\mu \xi_j^\mu.$$

The stabilization parameter is given by

$$\begin{aligned} x_i^\mu &= \xi_i^\mu h_i(I^\mu) = \sum_j J_{ij}(\mu) \xi_i^\mu \xi_j^\mu \\ &= \xi_i^\mu \left[\frac{\xi_i^\mu \sum_j (\xi_j^\mu)^2}{N} + \sum_j J_{ij}(\mu - 1) \xi_j^\mu \right] = 1 + x_i^{(\mu)}, \end{aligned}$$

where $x_i^{(\mu)} = \xi_i^\mu h_i^{(\mu)}$ with $h_i^{(\mu)} = \sum_j J_{ij}(\mu - 1) \xi_j^\mu$.

The pattern I^μ would be well memorized if we managed to make the noise term $x_i^{(\mu)}$ vanish. This can be achieved by introducing an extra contribution,

$$-\frac{\xi_j^\mu}{N} \sum_k J_{ik}(\mu - 1) \xi_k^\mu,$$

into the learning dynamics. This extra term exactly cancels out the noisy field at every learning step. The learning rule, called the Widrow Hoff learning rule, becomes (Fig. 7.4)

$$J_{ij}(\mu) = J_{ij}(\mu - 1) + \frac{1}{N} \left[\xi_i^\mu - \sum_k J_{ik}(\mu - 1) \xi_k^\mu \right] \xi_j^\mu. \quad (7.5)$$

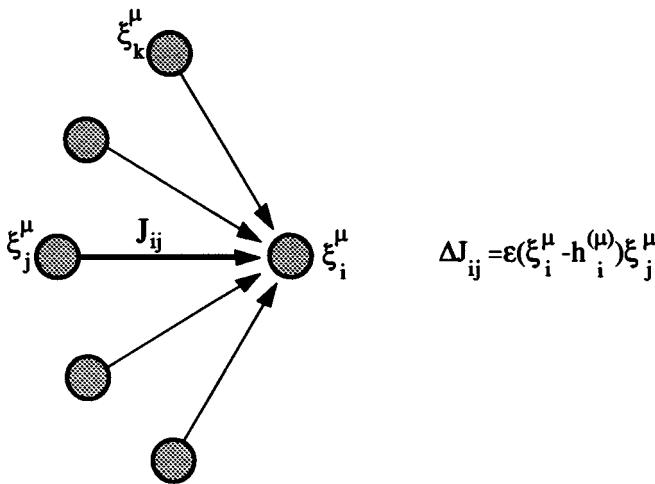


Figure 7.4. The Widrow Hoff learning rule.

To show that the noise term is canceled out, we compute the local field on neuron i :

$$\begin{aligned} h_i(I^\mu) &= \sum_j J_{ij}(\mu) \xi_j^\mu \\ &= \sum_j J_{ij}(\mu-1) \xi_j^\mu + \xi_i^\mu - \frac{1}{N} \sum_j (\xi_j^\mu)^2 \sum_k J_{ik}(\mu-1) \xi_k^\mu \\ &= \xi_i^\mu + \sum_j J_{ij}(\mu-1) \xi_j^\mu - \sum_k J_{ik}(\mu-1) \xi_k^\mu = \xi_i^\mu \end{aligned}$$

and therefore $x_i^\mu = 1$.

It is worth noting that the algorithm is asymmetrical in character.

Remarks

- a) The Widrow Hoff rule (7.5) is similar to the rule proposed by Rescorla and Wagner to account for context effects in classical conditioning experiments (see section 2.4). It has been rediscovered several times and introduced along the lines displayed in this section by Kinzel.
- b) The Widrow Hoff learning rule can be rewritten in the following form:

$$\Delta J_{ij}(\mu) = \frac{1}{N} f_{ij}(\mu) \xi_i^\mu \xi_j^\mu,$$

with $f_{ij}(\mu) = 1 - x_i^{(\mu)}$, $x_i^{(\mu)} = \sum_k J_{ik}(\mu-1) \xi_i^\mu \xi_k^\mu$.

That is to say, it is similar to a Hebbian rule with a prefactor which decreases as the stabilizing parameter x_i^μ increases.

c) The unit matrix $\mathbf{I}^{(N)}$, $(\mathbf{I}^{(N)})_{ij} = \delta_{ij}$ is always a fixed point of the algorithm (7.5) whatever the number P of memorized patterns. In the limit of very large N ’s the connectivity matrix becomes the unit matrix, which means that all patterns are stable but that the radii of basins of attraction are all zero. This artefact is avoided by setting the diagonal terms to zero during the learning session. The complete learning rule is therefore

$$\Delta J_{ij}(\mu) = \frac{1}{N} \left[\xi_i^\mu - \sum_k J_{ik}(\mu - 1) \xi_k^\mu \right] \xi_j^\mu, \quad J_{ii} = 0. \quad (7.6)$$

The same effect arises in pure Hebbian learning algorithms: with

$$J_{ij} = \frac{1}{N} \sum_\mu \xi_i^\mu \xi_j^\mu,$$

the local field, for pattern I^1 say, is given by

$$\begin{aligned} h_i(I^1) &= \frac{1}{N} \sum_{j=1}^N \sum_{\mu=1}^P \xi_i^\mu \xi_j^\mu \xi_j^1 \\ &= \frac{P}{N} \xi_i^1 + \frac{1}{N} \sum_{j \neq i} \xi_i^1 \xi_j^1 \xi_j^1 + \frac{1}{N} \sum_{j \neq i} \sum_{\mu \neq 1} \xi_i^\mu \xi_j^\mu \xi_j^1 \\ &= (\alpha + 1) \xi_i^1 + \frac{1}{N} \sum_{j \neq i}^N \sum_{\mu \neq 1}^P (\pm 1) = (\alpha + 1) \xi_i^1 + O(\alpha^{1/2}). \end{aligned}$$

Since the $1 + \alpha$ term always dominates the noise term $\sqrt{\alpha}$, any configuration is stable, whatever $\alpha = P/N$. The size of basins of attraction, however, is reduced to a single state and all memory properties are lost. This is avoided by compelling the self-connection terms to be zero: $J_{ii} \equiv 0$ (see Eq. 4.2).

d) If one stores a string of different patterns, the pattern I^μ which is learned last is well imprinted but nothing guarantees that the noise terms $x_i^{(\mu')}$ associated with formerly imprinted patterns, $\mu' < \mu$, remain equal to zero. In actual fact they don’t, and the patterns imprinted in the past are progressively erased from memory. Therefore this algorithm also causes a palimpsest effect. On the other hand, instead of learning infinite strings of patterns, it is also possible to learn a finite set of P configurations by recycling the patterns again and again. It will be

proved in the next section that the process converges and that the maximum memory storage capacity is $\alpha_c = P_c/N = 1$. It will be also observed that in the limit $\alpha \rightarrow 1$ the basins of attraction are vanishingly small.

7.3.2 Synaptic efficacies as projection matrices

In the N -dimensional phase space of neuronal states the set of memorized patterns $I^\mu = \{\xi_i^\mu = +/-1\}$, $\mu = 1, 2, \dots, P$ spans a subspace \mathcal{E}^{\parallel} whose dimension P^{\parallel} is less or equal to P . It is less when the patterns are non-linearly independent. We call \mathcal{E}^{\perp} the complementary subspace. A pattern I^μ is stable if the local fields $h_i(I^\mu)$, $i = 1, 2, \dots, N$ all align along the components ξ_i^μ of the pattern. This can be written as

$$\sum_j J_{ij} \xi_j^\mu = \lambda_i^\mu \xi_i^\mu, \quad \text{with } \lambda_i^\mu > 0. \quad (7.7)$$

This formula is quite general. In projection algorithms one introduces constraints on constants λ_i^μ in particular one chooses to make the constants site-independent. The equations (7.7) become

$$\sum_j J_{ij} \xi_j^\mu = \lambda^\mu \xi_i^\mu, \quad \text{with } \lambda^\mu > 0, \quad (7.8)$$

which show that the P patterns $I^\mu = \tilde{\xi}^\mu$ must be eigenvectors of the $N \times N$ matrix \mathbf{J} of interactions with positive eigenvalues λ^μ :

$$\mathbf{J} \cdot \tilde{\xi}^\mu = \lambda^\mu \tilde{\xi}^\mu. \quad (7.9)$$

The problem is to find \mathbf{J} . The patterns I^μ play the same role. One therefore assumes that the eigenvalues are pattern-independent. Then a harmless renormalization of \mathbf{J} allows us to make them all equal to 1.

$$\mathbf{J} \cdot \tilde{\xi}^\mu = \tilde{\xi}^\mu. \quad (7.10)$$

According to Eq. (7.9) (and all the more so to Eq. (7.10)), \mathbf{J} leaves the subspace \mathcal{E}^{\parallel} unchanged. This condition is not enough: one also has to hinder the stabilization of vectors $\tilde{\xi}^\perp$ of the complementary subspace \mathcal{E}^{\perp} , since $\tilde{\xi}^\perp$, a state which is not to be stabilized, would be stabilized anyway. To cure this defect and make all states $\tilde{\xi}^\perp$ unstable, one endows the complementary subspace \mathcal{E}^{\perp} with a basis of vectors $\tilde{\xi}^\perp$ which are eigenvectors of J with non-positive eigenvalues. A possible choice is to assume that all these eigenvalues vanish. One has

$$\mathbf{J} \cdot \tilde{\xi}^\nu = \lambda^\nu \tilde{\xi}^\nu,$$

with $\lambda^\nu = 1$, $\nu = 1, \dots, P^{\parallel}$ and $\lambda^\nu = 0$, $\nu = P^{\parallel} + 1, \dots, N$, and \mathbf{J} transforms any vector of the N -dimensional phase into its projection in the \mathcal{E}^{\parallel} space. The matrix of interactions is therefore given by

$$\mathbf{J} = \mathcal{P}^{\parallel}, \quad (7.11)$$

where \mathcal{P}^{\parallel} is the projection operator into space \mathcal{E}^{\parallel} .

7.3.3 Learning dynamics

In this section we look for an iterative process which progressively builds the projection operator \mathcal{P}^{\parallel} and therefore the interaction matrix J . The idea is to find a formula which relates $\mathcal{P}_{\mu}^{\parallel}$, which projects into the space spanned by μ patterns, to $\mathcal{P}_{\mu-1}^{\parallel}$, the operator which projects into the space spanned by $(\mu - 1)$ patterns. This can be achieved by using the Schmidt orthogonalization procedure. Let us start from the first pattern,

$$\tilde{\xi}_1^{\perp} = \tilde{\xi}_1.$$

One builds a vector $\tilde{\xi}_2^{\perp}$ orthogonal to the first pattern by subtracting from $\tilde{\xi}_2$ its projection on $\tilde{\xi}_1$. The projection is given by

$$\frac{\tilde{\xi}_1^{\perp}}{\|\tilde{\xi}_1^{\perp}\|} \frac{\tilde{\xi}_1^{\perp T} \cdot \tilde{\xi}_2}{\|\tilde{\xi}_1^{\perp}\|},$$

whence
$$\tilde{\xi}_2^{\perp} = \tilde{\xi}_2 - \tilde{\xi}_1^{\perp} \left(\frac{\tilde{\xi}_1^{\perp T} \cdot \tilde{\xi}_2}{\|\tilde{\xi}_1^{\perp}\|^2} \right).$$

$\tilde{\xi}^T$ is the transpose (line) vector of the (column) vector $\tilde{\xi}$. The process is repeated and after the μ th pattern one finds

$$\tilde{\xi}_{\mu}^{\perp} = \tilde{\xi}_{\mu} - \sum_{\nu=1}^{\mu-1} \tilde{\xi}_{\nu}^{\perp} \left(\frac{\tilde{\xi}_{\nu}^{\perp T} \cdot \tilde{\xi}_{\mu}}{\|\tilde{\xi}_{\nu}^{\perp}\|^2} \right),$$

which can be written

$$\tilde{\xi}_{\mu}^{\perp} = \mathcal{P}_{\mu-1}^{\perp} \cdot \tilde{\xi}_{\mu},$$

with
$$\mathcal{P}_{\mu-1}^{\perp} = I^{(N)} - \sum_{\nu=1}^{\mu-1} \frac{\tilde{\xi}_{\nu}^{\perp} \cdot \tilde{\xi}_{\nu}^{\perp T}}{\|\tilde{\xi}_{\nu}^{\perp}\|^2}.$$

$\mathbf{I}^{(N)}$ is the N -dimensional unit matrix. If one singles out the last term

of the series, one obtains a recursion relation between the projectors \mathcal{P}_μ^\perp :

$$\begin{aligned}\mathcal{P}_\mu^\perp &= \mathbf{I}^{(N)} - \sum_{\nu=1}^{\mu} \frac{\tilde{\xi}_\nu^\perp \cdot \tilde{\xi}_\nu^{\perp T}}{\|\tilde{\xi}_\nu^\perp\|^2} \\ &= \mathcal{P}_{\mu-1}^\perp - \frac{\tilde{\xi}_\mu^\perp \cdot \tilde{\xi}_\mu^{\perp T}}{\|\tilde{\xi}_\mu^\perp\|^2} \\ &= \mathcal{P}_{\mu-1}^\perp - \frac{(\mathcal{P}_{\mu-1}^\perp \cdot \tilde{\xi}_\mu) \cdot (\tilde{\xi}_\mu^T \cdot \mathcal{P}_{\mu-1}^\perp)}{\|\mathcal{P}_{\mu-1}^\perp \cdot \tilde{\xi}_\mu\|^2};\end{aligned}$$

and therefore, using $\mathcal{P}^\parallel = \mathbf{I}^{(N)} - \mathcal{P}^\perp$,

$$\mathcal{P}_\mu^\parallel = \mathcal{P}_{\mu-1}^\parallel + \frac{(\mathbf{I}^{(N)} - \mathcal{P}_{\mu-1}^\parallel) \cdot \tilde{\xi}_\mu \cdot \tilde{\xi}_\mu^T \cdot (\mathbf{I}^{(N)} - \mathcal{P}_{\mu-1}^\parallel)}{\|(\mathbf{I}^{(N)} - \mathcal{P}_{\mu-1}^\parallel) \cdot \tilde{\xi}_\mu\|^2}.$$

Since the operator $\mathcal{P}_\mu^\parallel$ is none other than the connectivity matrix $\mathbf{J}(\mu)$, the learning algorithm can be written as

$$\begin{aligned}\Delta J_{ij} &= J_{ij}(\mu) - J_{ij}(\mu-1) \\ &= \frac{[\xi_i^\mu - \sum_k J_{ik}(\mu-1) \xi_k^\mu] [\xi_j^\mu - \sum_k J_{jk}(\mu-1) \xi_k^\mu]}{\sum_k [\xi_k^\mu - \sum_l J_{kl}(\mu-1) \xi_l^\mu]^2},\end{aligned}$$

whence the projection algorithm

$$\Delta J_{ij} = \frac{(\xi_i^\mu - h_i^{(\mu)}) (\xi_j^\mu - h_j^{(\mu)})}{\sum_k (\xi_k^\mu - h_k^{(\mu)})^2}, \quad (7.12)$$

with $h_i^{(\mu)} = \sum_{j=1}^N J_{ij}(\mu-1) \xi_j^\mu$.

This algorithm is very efficient. It does not involve any matrix inversion, and since the convenient projector is found at every step, it stops after P iterations. Moreover, it is not necessary to worry about the linear independence of the patterns. Indeed, if a new pattern is a linear combination of already imprinted patterns the corresponding step will be simply ignored.

7.3.4 An explicit expression for the projection operator

Let us define an $N \times P$ matrix Σ as the matrix whose columns are the P patterns $\tilde{\xi}^\mu$ to be memorized:

$$\Sigma = \begin{pmatrix} \xi_1^1 & \xi_1^2 & \cdots & \cdots & \xi_1^P \\ \xi_2^1 & \xi_2^2 & \cdots & \cdots & \xi_2^P \\ \vdots & \vdots & & \ddots & \vdots \\ \xi_N^1 & \xi_N^2 & \cdots & \cdots & \xi_N^P \end{pmatrix}.$$

If $P < N$ and if the P patterns are linearly independant but not necessarily orthogonal, the projection matrix \mathcal{P}^{\parallel} is given by

$$\mathcal{P}^{\parallel} = \Sigma \cdot (\Sigma^T \cdot \Sigma)^{-1} \cdot \Sigma^T. \quad (7.13)$$

Σ^T is the $P \times N$ transpose matrix of Σ . The formula (7.13) is proved by verifying that

$$\mathcal{P}^{\parallel} \cdot \Sigma = \Sigma \cdot (\Sigma^T \cdot \Sigma)^{-1} \cdot (\Sigma^T \cdot \Sigma) = \Sigma,$$

which shows that

$$\mathcal{P}^{\parallel} \cdot \tilde{\xi}^{\mu} = \tilde{\xi}^{\mu} \quad \forall \mu \quad \text{and} \quad \mathcal{P}^{\parallel} \cdot \tilde{\xi}^{\parallel} = \tilde{\xi}^{\parallel}$$

for any linear combination $\tilde{\xi}^{\parallel}$ of pattern vectors $\tilde{\xi}^{\mu}$. Moreover, any vector $\tilde{\xi}^{\perp}$ of the complementary space is such that $\Sigma^T \cdot \tilde{\xi}^{\perp} = 0$ and therefore $\mathcal{P}^{\parallel} \cdot \tilde{\xi}^{\perp} = 0$, which proves that \mathcal{P}^{\parallel} is the projection operator on \mathcal{E}^{\parallel} indeed.

Remarks

a) If $P < N$ and if the patterns are orthonormal the projection matrix becomes

$$\mathcal{P}^{\parallel} = \Sigma \cdot \Sigma^T, \quad (7.14)$$

which is the usual Hebbian rule,

$$\mathcal{P}_{ij}^{\parallel} = J_{ij} = \sum_{\mu} \xi_i^{\mu} \xi_j^{\mu}.$$

In actual fact there is a normalization constant $1/N$ in the Hebbian rule. It can be recovered by setting $\lambda = 1/N$ in Eq. (7.9) instead of $\lambda = 1$.

b) It is worth noting that the matrix (7.13) can be written as

$$\mathcal{P}^{\parallel} = \Sigma \cdot \Gamma^{-1} \cdot \Sigma^T, \quad (7.15)$$

$$\text{where} \quad \Gamma = \Sigma^T \cdot \Sigma, \quad (\Gamma)_{\mu\mu'} = \sum_i \xi_i^{\mu} \xi_i^{\mu'}$$

is the $P \times P$ correlation matrix between the patterns I^{μ} . The matrix elements of (7.15) are given by

$$J_{ij} = \sum_{\mu} \sum_{\mu'} \xi_i^{\mu} (\Gamma^{-1})_{\mu\mu'} \xi_j^{\mu'}. \quad (7.16)$$

One notes that, owing to the symmetry of the correlation matrix Γ , the interactions are symmetrical, $J_{ij} = J_{ji}$, whereas the learning dynamics are not. The formula (7.16) is valid even if the patterns are non-orthogonal, but they have to be linearly independent, otherwise the correlation matrix Γ cannot be inverted. If such a case arises a simple solution is to forget the supernumerary patterns. For example if $P > N$, the patterns are necessarily dependent and the operator \mathcal{P}^{\parallel} has to be built using $P = N$ patterns. Then $\mathcal{P}^{\parallel} = \mathbf{I}^{(N)}$ and all configurations are stable. This shows that the memory storage capacity associated with projection matrices is limited to $P_c \leq N$ and that the radius of basins of attraction, if such a limit is reached, is zero.

c) Following Kanter and Sompolinsky, let us check the size of the basins of attraction determined by the projection matrix \mathcal{P}^{\parallel} . Starting from pattern I^{μ} , one defines a new pattern $I^{\mu,1}$ whose neuronal states are parallel to those of I^{μ} , $\sigma_i = \xi_i^{\mu}$ except for neuron 1, $\sigma_1 = -\xi_1^{\mu}$. Then one checks whether the local field tends to flip σ_1 in the direction of ξ_1^{μ} . If it fails to do so the size of the basin is zero. The local field on neuron 1 is given by

$$h_1(I^{\mu,1}) = \sum_{j \neq 1} J_{1j} \xi_j^{\mu} + J_{11} \sigma_1 = \sum_{j \neq 1} J_{1j} \xi_j^{\mu} - J_{11} \xi_1^{\mu}.$$

Since \mathbf{J} is a projection matrix one has

$$\sum_j J_{1j} \xi_j^{\mu} = \xi_1^{\mu} = \sum_{j \neq 1} J_{1j} \xi_j^{\mu} + J_{11} \xi_1^{\mu}$$

and therefore
$$h_1(I^{\mu,1}) = (1 - 2J_{11}) \xi_1^{\mu}. \quad (7.17)$$

On the other hand, the average diagonal term J_{11} is derived from the following formulae:

$$\begin{aligned} \sum_i J_{ii} &= \sum_i \sum_{\mu\mu'} \xi_i^{\mu} (\Gamma^{-1})_{\mu\mu'} \xi_i^{\mu'} \\ &= \sum_{\mu\mu'} (\Gamma^{-1})_{\mu\mu'} (\Gamma)_{\mu\mu'} = \sum_{\mu} 1 = P \end{aligned}$$

and

$$\sum_i J_{ii} \simeq N J_{ii},$$

whence

$$J_{11} \simeq \frac{P}{N} = \alpha.$$

Finally, the local is given by

$$h_1(I^{\mu,1}) = (1 - 2\alpha)\xi_1^\mu$$

and one sees that the neuron 1 is destabilized if

$$\alpha > \frac{1}{2} = \alpha_c, \quad \text{that is,} \quad P_c = \frac{1}{2}N.$$

To improve the capacity one can set $J_{11} = 0$ in Eq. (7.17). Then the local field on site 1 is given by

$$h_1(I^{\mu,1}) = \xi_1^\mu \quad (7.18)$$

and the state σ_1 is stable at least as long as $P < N$, which is the largest dimension of \mathcal{E}^{\parallel} . One notes that Eq. (7.18) is the equation for the local fields in the Widrow Hoff algorithm.

One sees how important is the necessity of cancelling the diagonal terms of \mathbf{J} . The reason is the following: since $\mathcal{P}_{ii}^{\parallel} = \alpha$, the final matrix can be written as

$$\mathbf{J} = \mathcal{P}^{\parallel} - \alpha \mathbf{I}^{(N)}. \quad (7.19)$$

On the other hand, we have defined \mathcal{P}^{\perp} ,

$$\mathcal{P}^{\perp} = \mathbf{I}^{(N)} - \mathcal{P}^{\parallel},$$

as the projector on the subspace \mathcal{E}^{\perp} perpendicular to \mathcal{E}^{\parallel} . A matrix \mathbf{J} given by

$$\mathbf{J} = \mathcal{P}^{\parallel} + \mathbf{B} \cdot \mathcal{P}^{\perp},$$

where \mathbf{B} is any $N \times N$ matrix, satisfies the stability conditions (7.9). But its eigenvalues $\lambda^\nu, \nu > P$ are not necessarily negative, which could lead to the stabilization of unwanted configurations. To guarantee the instability of the complementary space one chooses

$$\mathbf{B} = -\lambda \mathbf{I}^{(N)}, \quad \lambda > 0.$$

The matrix of connections becomes

$$\begin{aligned} \mathbf{J} &= \mathcal{P}^{\parallel} - \mathcal{P}^{\perp} = \mathcal{P}^{\parallel} - \lambda(\mathbf{I}^{(N)} - \mathcal{P}^{\parallel}) \\ &= (1 + \lambda)\left(\mathcal{P}^{\parallel} - \frac{\lambda}{1 + \lambda}\mathbf{I}^{(N)}\right). \end{aligned} \quad (7.20)$$

The expressions (7.19) and (7.20) can be made identical by setting

$$\lambda = \frac{\alpha}{1 - \alpha},$$

which is positive for $\alpha < 1$. The increase of the memory capacity from $\alpha = \frac{1}{2}$ to $\alpha = 1$ is related to the stabilizing effect of negative eigenvalues of $\mathbf{B} \cdot \mathcal{P}^\perp$.

7.3.5 Other projection algorithms

The iterative formula (7.12) suggests several learning rules (Dreyfus).

Neglecting the membrane potential contribution in the denominator, one finds a symmetrical learning rule:

$$\Delta J_{ij} = \frac{1}{N} (\xi_i^\mu - h_i^{(\mu)}) (\xi_j^\mu - h_j^{(\mu)}).$$

Then one can neglect either the postsynaptic membrane potential,

$$\Delta J_{ij} = \frac{1}{N} \xi_i^\mu (\xi_j^\mu - h_j^{(\mu)}),$$

or the presynaptic membrane potential,

$$\Delta J_{ij} = \frac{1}{N} \xi_j^\mu (\xi_i^\mu - h_i^{(\mu)}), \quad \text{with} \quad h_i^{(\mu)} = \sum_k J_{ik}(\mu - 1) \xi_k^\mu,$$

which is the Widrow Hoff rule. Finally the Hebbian rule is recovered by neglecting both membrane potentials $h_i^{(\mu)}$ and $h_j^{(\mu)}$.

If the set of patterns I^μ is learned iteratively, all but the Hebbian rule have definite fixed points \mathbf{J}^* , which are given by

$$\mathbf{J}^* = \mathcal{P}^\parallel + \mathbf{B} \cdot \mathcal{P}^\perp$$

where the matrix \mathbf{B} depends on initial conditions. One has

$$\tilde{h}^{\mu*} = \mathbf{J}^* \cdot \tilde{\xi}^\mu = \mathcal{P}^\parallel \cdot \tilde{\xi}^\mu + \mathbf{B} \cdot \mathcal{P}^\perp \cdot \tilde{\xi}^\mu = \tilde{\xi}^\mu + \mathbf{B} \cdot \tilde{0} = \tilde{\xi}^\mu,$$

that is, $h_i^{\mu*} = \xi_i^\mu$ for all sites i and therefore

$$\Delta J_{ij}^* = 0$$

for the three rules. Let $J(0)$ be the initial matrix and let $\mathbf{J}(\mu - 1)$ be the matrix after step $(\mu - 1)$ has been completed. One writes

$$\begin{aligned} \mathbf{J}(\mu - 1) &= \mathbf{I}^{(N)} \cdot \mathbf{J}(\mu - 1) = (\mathcal{P}_\mu^\parallel + \mathcal{P}_\mu^\perp) \cdot \mathbf{J}(\mu - 1) \\ &= \mathbf{J}^\parallel(\mu - 1) + \mathbf{J}^\perp(\mu - 1). \end{aligned}$$

The Widrow Hoff rule, for example, can be cast in the following form:

$$\Delta J_{ij}(\mu) = \frac{1}{N} \mathbf{P}_{ij}^\mu \left[1 - (\mathbf{J}(\mu-1) \cdot \mathbf{P}^\mu)_{ii} \right] = \Delta J_{ij}^{\parallel}(\mu) + \Delta J_{ij}^{\perp}(\mu),$$

where

$$\mathbf{P}^\mu = \tilde{\xi}^\mu \cdot \tilde{\xi}^{\mu T}$$

is the elementary projection operator in the one-dimensional subspace spanned by $\tilde{\xi}^\mu$. Therefore

$$\mathbf{J}^\perp(\mu-1) \cdot \mathbf{P}^\mu = 0$$

and $\Delta J_{ij}^{\perp}(\mu) = 0$, which shows that \mathbf{J}^\perp does not change during the iteration. The fixed point is then given by

$$\mathbf{J}^* = \mathcal{P}^{\parallel} + \mathbf{J}(0) \cdot \mathcal{P}^{\perp}. \quad (7.21)$$

As above, the cancelling of self-interactions must be supplemented to the rule (7.21). For the proof to be complete it is necessary to show that the solution (7.21) is unique. We shall see in section 7.5 that the space of \mathbf{J} explored by the projection algorithms is convex and that therefore the solution is unique.

7.3.6 A brief survey on the history of projection algorithms

The technique of projection algorithms has been introduced in the theory of neural networks by Kohonen. He was inspired by the works of Penrose on pseudo-inverse matrices (see below). Kohonen has applied the algorithm to bi-layered neural networks. It is clear that for the projection algorithms to be applicable, the neuronal architecture must comprise no hidden units. The problem is to associate P patterns $I^{\mu, \text{in}} = \{\xi_j^{\mu, \text{in}}\}$, $j = 1, \dots, N^I$, where N^I is the number of units of the input layer, with P patterns $I^{\mu, \text{out}} = \{\xi_i^{\mu, \text{out}}\}$, $i = 1, \dots, N^O$, where N^O is the number of units of the output layer. The states of the output layer are given by

$$\sigma_i^{\mu, \text{out}} = \text{sign}[h_i(I^{\mu, \text{in}})]$$

and therefore the association is achieved if

$$\text{sign}\left(\sum_j J_{ij} \xi_j^{\mu, \text{in}}\right) = \xi_i^{\mu, \text{out}}$$

There is no memory in such a network since there is no feedback. This system, sometimes called a hetero-associative network, can model sensory or effector networks transforming the internal representations of the central nervous system into effective motions for example. With

the P input patterns one builds an $N^I \times P$ matrix Σ^{in} . Similarly with the P output patterns one builds an $N^O \times P$ matrix Σ^{out} . The solution of equation

$$\mathbf{J} \cdot \Sigma^{\text{in}} = \Sigma^{\text{out}}, \quad \text{is given by } \mathbf{J} = \Sigma^{\text{out}} \cdot (\Sigma^{\text{in}})^+,$$

where $(\Sigma^{\text{in}})^+$ is the pseudo-inverse matrix of Σ^{in} (see below).

The application of the projection matrix to fully connected networks, where $\Sigma^{\text{in}} \equiv \Sigma^{\text{out}}$, is due to Dreyfus, Personnaz and Guyon. Since the interactions given by the projection algorithm are symmetrical it is possible to use the tools of statistical mechanics we introduced in the last chapter to find analytical results. Calculations appealing to the replica technique have been carried out by Sompolinsky *et al.*. Among other results their derivation confirmed an upper memory storage capacity of $\alpha_c = 1$. This is far better than the capacity of the pure Hebbian model which is $\alpha_c = 0.14$ but the algorithm is still not optimal since, according to the results of the last section, the theoretical limit for uncorrelated patterns is $\alpha_c = 2$.

A few properties of pseudo-inverse matrices

The $N \times M$ matrix

$$\Sigma^+ = (\Sigma^T \cdot \Sigma)^{-1} \cdot \Sigma^T$$

is called *the pseudo-inverse matrix* of Σ . It has the following properties:

$$\Sigma \cdot \Sigma^+ \cdot \Sigma = \Sigma, \quad \Sigma^+ \cdot \Sigma \cdot \Sigma^+ = \Sigma^+, \quad \Sigma^+ \cdot \Sigma = \mathbf{I}^{(P)}, \quad (\Sigma \cdot \Sigma^+)^T = \Sigma \cdot \Sigma^+.$$

Pseudo-inverse matrices are related to mean square analysis. Let us assume that we want to solve the set of linear equations

$$\tilde{\mathbf{y}} = \mathbf{A} \cdot \tilde{\mathbf{x}},$$

where \mathbf{A} is a $N \times M$ matrix with $N > M$. $\tilde{\mathbf{y}}$ is a $N \times 1$ vector and $\tilde{\mathbf{x}}$ is a $M \times 1$ vector. Therefore there does not exist a solution $\tilde{\mathbf{x}}$ in general. Let us look for the best solution anyway, the solution which minimizes the norm

$$\| \tilde{\mathbf{y}} - \mathbf{A} \cdot \tilde{\mathbf{x}} \|^2 = (\tilde{\mathbf{y}} - \mathbf{A} \cdot \tilde{\mathbf{x}})^T \cdot (\tilde{\mathbf{y}} - \mathbf{A} \cdot \tilde{\mathbf{x}}).$$

Differentiating this expression one finds $2(\tilde{\mathbf{y}} - \mathbf{A} \cdot \tilde{\mathbf{x}})^T \cdot \mathbf{A} \cdot \delta \tilde{\mathbf{x}} = 0$ for all $\delta \tilde{\mathbf{x}}$. That is, $\tilde{\mathbf{y}}^T \cdot \mathbf{A} = \tilde{\mathbf{x}}^T \cdot \mathbf{A}^T \cdot \mathbf{A}$, and finally

$$\tilde{\mathbf{x}} = \mathbf{A}^+ \cdot \tilde{\mathbf{y}}, \quad \mathbf{A}^+ = (\mathbf{A}^T \cdot \mathbf{A})^{-1} \cdot \mathbf{A}^T.$$

Therefore the best solution is given by the pseudo-inverse matrix. The discovery of pseudo-inverse matrices is due to Penrose.

7.4 The perceptron learning rules

The projection learning algorithm makes the memory storage capacity increase from $P_c = 0.14 \times N$, the capacity of simple Hebbian networks, to $P_c = N$. This is a substantial improvement but we have seen that neural networks are able to store up to $P_c = 2 \times N$ random patterns. In this paragraph we present a learning dynamics, namely the perceptron algorithm, which enables the network to reach this limit. There is a rather high price to pay however. Contrary to projection algorithms, closed expressions of the fixed points $\tilde{\mathbf{J}}^*$ are not known for the perceptron algorithm, which makes very difficult the analytical study of its properties.

7.4.1 Introducing cost functions in learning dynamics

In Chapter 3, the study of the dynamics of neural states has been made easy by the introduction of Lyapunov and energy functions whose main property is that they are non-increasing functions of time. It is tempting to appeal to the same technique in the theory of learning and to define a *cost function* $H(\{J_{ij}\})$ of synaptic efficacies which the learning dynamics strives to make smaller and smaller.

Once the cost function is defined, the simplest mechanism which makes certain that it decreases during the learning stage is the *gradient algorithm*: the synaptic efficacies are modified in proportion to the corresponding partial derivatives of the cost function:

$$\Delta J_{ij} = -\varepsilon \frac{\partial H}{\partial J_{ij}}, \quad \varepsilon > 0. \quad (7.22)$$

There is no restriction on the symmetry of synaptic efficacies and the dynamics generally yields asymmetric synaptic matrices $\tilde{\mathbf{J}}$. H is a decreasing function of time since

$$\delta H = \sum_{ij} \frac{\partial H}{\partial J_{ij}} \Delta J_{ij} = -\varepsilon \sum_{ij} \left(\frac{\partial H}{\partial J_{ij}} \right)^2 < 0.$$

The parameter ε must be small enough for the trajectory to follow the meanderings of the cost function, but it must be large enough if one wants the algorithm to reach a fixed point of $\tilde{\mathbf{J}}$ in not too long a time. The central problem, however, is to determine the cost function. Its solution fully depends on the idea we have regarding the way neural networks process information. If one considers that the role of learning is to make the responses of the network to given stimuli as close as possible to desired outputs, the simplest idea is to identify the cost function with an *error function*.

An error function tells how far from the desired output $I^{\mu, \text{out}} = \{\xi_{i \in \mathcal{O}}^\mu\}$ is the observed output $\{\sigma_{i \in \mathcal{O}}(I^{\mu, \text{in}})\}$. It is given by

$$H(I^\mu) = \sum_{i \in \mathcal{O}} (\sigma_i(I^{\mu, \text{in}}) - \xi_i^\mu)^2. \quad (7.23)$$

This quantity is directly proportional to the number of states that are dissimilar in the desired and the observed outputs. This amounts to saying that H is proportional to the Hamming distance between the two sorts of outputs. The Hamming distance $d^{\mu\mu'}$ between two states I^μ and $I^{\mu'}$ is the number of states which are different in I^μ and $I^{\mu'}$. The Hamming distance is related to the scalar product $I^\mu \cdot I^{\mu'}$ by

$$d^{\mu\mu'} = \frac{1}{2}(N - I^\mu \cdot I^{\mu'}). \quad (7.24)$$

The cost defined in Eq. (7.23) varies with time since, owing to the existence of noise and transient behaviors, the output states $\sigma_i(I^{\mu, \text{in}})$ (written σ_i^μ for short) are fluctuating quantities. Since the learning dynamics is generally considered as being slow compared with the neural states dynamics, the effects of transients may be neglected and the output may be thermally averaged:

$$H(I^\mu) = \sum_{i \in \mathcal{O}} (\langle \sigma_i^\mu \rangle - \xi_i^\mu)^2. \quad (7.25)$$

Equation (7.25) determines a ‘local’ error function in the sense that the cost is defined for a given pattern I^μ . This is the sort of cost function one uses in serial learning dynamics where the patterns to be memorized are learned one after the other. Parallel learning dynamic strives to minimize the following ‘global’ error function:

$$H = \sum_{\mu=1}^P \sum_{i \in \mathcal{O}} (\langle \sigma_i^\mu \rangle - \xi_i^\mu)^2. \quad (7.26)$$

In parallel learning dynamics the synaptic efficacies are modified so as to stabilize all patterns at every learning step. Parallel learning dynamics is a procedure which generally proves to be better than serial dynamics.

There are other ways of building cost functions. One may consider that the system strives to make the patterns to be memorized as stable as possible. This amounts to saying that the role of learning dynamics is to increase the values of stabilization parameters x_i^μ :

$$x_i^\mu = \xi_i^\mu h_i(I^\mu) = \sum_j J_{ij} \xi_i^\mu \xi_j^\mu.$$

The cost function is then given by

$$H(I^\mu) = \sum_i g(x_i^\mu) \quad (7.27)$$

for serial learning and by

$$H = \sum_\mu \sum_i g(x_i^\mu) \quad (7.28)$$

for parallel learning. The only constraint on $g(x)$ is that it is a monotonous decreasing function; the larger is x the smaller is H .

Hamming distances or stabilization parameters are not the sole criteria which can be used in the building of learning dynamics. Other quantities such as the relative entropy or the asymmetry parameter η of the interaction matrix $\tilde{\mathbf{J}}$ may prove to be determinant as well. The relative entropy is the cost function which determines the learning dynamics of the Boltzmann machine (see section 8.2.3 below). As one can see, the problem of learning remains open.

The associative learning rules we have seen so far may be derived from special cost functions. For example let us take the sum of stabilization parameters x_i^μ as the cost function ($g(x) = -x$):

$$H = - \sum_\mu \sum_i x_i^\mu = - \sum_\mu \sum_{i,j} J_{ij} \xi_i^\mu \xi_j^\mu.$$

The gradient dynamics yields $J_{ij} = \varepsilon \sum_\mu \xi_i^\mu \xi_j^\mu$, which is the Hebbian rule.

As an example of a learning rule based upon the error correction paradigm, we consider the case of the *perceptron architecture* which we introduced at the beginning of this chapter (see Fig. 4.1). It is recalled that this is a simple system comprising N^I input neurons j and a single output unit whose state is σ .

The cost function is the error function (7.23)

$$H(I^\mu) = (\langle \sigma^\mu \rangle - \xi^\mu)^2, \quad (7.29)$$

where, according to the results of Chapter 3, the thermal average is given by

$$\langle \sigma^\mu \rangle = S \left(\beta \sum_{j \in I} J_j \xi_j^{\mu, \text{in}} \right)$$

In Adaline (an adaptive linear network), the machine put forward and built by Widrow in 1961, it is assumed that β is small enough for the system to function in its linear regime (Fig. 7.5). Then

$$H(I^\mu) = \left(\xi^\mu - \beta \sum_j J_j \xi_j^{\mu, \text{in}} \right)^2$$

and the gradient dynamics yields

$$J_j(\mu) = \varepsilon' \left(\xi^\mu - \beta \sum_j J_j \xi_j^\mu \right) \xi_j^{\mu, \text{in}}, \quad (7.30)$$

with $\varepsilon' = \varepsilon \beta$. One recovers the Widrow Hoff (Rescorla Wagner) rule. By setting $\beta = 1$ one also recovers one of the projection algorithms we studied in section 7.3. At the time Adaline was created, cybernetics was the main source of inspiration and the last term in Eq. (7.30) may be thought of as a counter-reaction term in a transducer machine and β as the gain of the counter-reaction loop. The concatenation of N^O Adaline networks, all sharing the same N^I input units, makes a bilayered system called Madaline (Multi-Adaline).

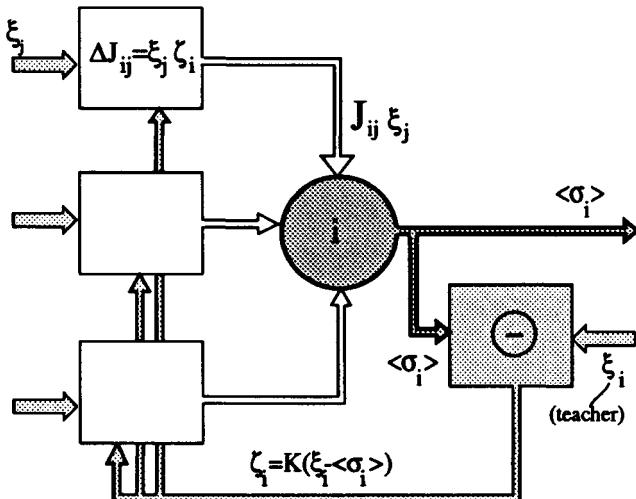


Figure 7.5. Adaline, a machine which learns (After Widrow).

7.4.2 The perceptron algorithm

Three years before Widrow created his Madaline, Rosenblatt devised a machine he called a perceptron. The machine of Rosenblatt was made of a layer of N^I binary input units j and a layer of N^O binary output units i . Every input unit was connected with every output unit and there were no interconnections between the output units i . This network is a superposition of N^O independent subsystems made of N^I input neurons j connected with one output neuron i . These are the subsystems which we have defined above as perceptrons.

Rosenblatt put forward a learning algorithm for his machine which is now known as the perceptron learning algorithm. This dynamics is a Hebbian rule which is supplemented by the following *perceptron learning principle*:

'Do not learn already stabilized patterns.'

Let $\tilde{\mathbf{J}} = \{J_j\}$ be the set of synaptic efficacies of a perceptron. The principle means that the interactions are not to be modified, $\Delta J_j = 0$, if the stabilization parameters are positive, $x^\mu > 0$. Otherwise the modification is Hebbian (see Fig. 7.6).

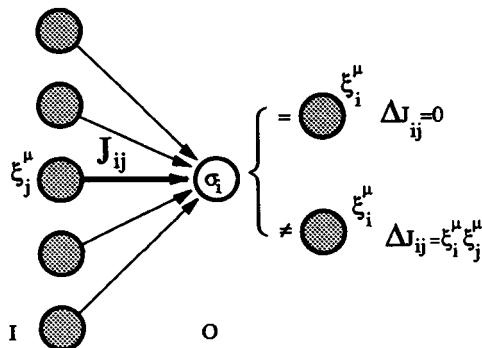


Figure 7.6. The perceptron learning rule.

Formally,

$$\Delta J_j(\mu) = f(\mu) \xi_j^\mu \xi_j^{\mu, \text{in}}, \quad (7.31)$$

with $f(\mu) = \frac{1}{2} [1 - \text{sign}(x^\mu)]$ and, as usual, $x^\mu = \sum_{j=0}^{N^I} J_j \xi_j^\mu \xi_j^{\mu, \text{in}}$.

The latter summation includes a $j = 0$ term. One recalls that the threshold θ of the neural dynamics can be materialized in the form of an extra input neuron labeled $j = 0$ whose state is always active $\xi_0 = +1$ and whose connection J_0 with the output neuron is

$$J_0 = -\theta.$$

This interaction is treated on an equal footing with respect to the other interactions J_j by the perceptron algorithm and is modified accordingly.

- The perceptron algorithm can be derived from a cost function defined by

$$H(I^\mu) = g\left(\sum_j J_j \xi_j^\mu \xi_j^\mu\right), \quad \text{with } g(x) = \begin{cases} -x & \text{if } x < 0, \\ 0 & \text{if } x > 0. \end{cases}$$

The suffix ‘in’ in $\xi_j^{\mu, \text{in}}$ is skipped for the sake of simplicity. The gradient dynamics Eq. (7.22) yields

$$\Delta J_j(\mu) = f(\mu) \xi^\mu \xi_j^\mu = f(x^\mu) \xi^\mu \xi_j^\mu,$$

where $f(x) = \frac{dg(x)}{dx} = \mathbf{1}(-x) \equiv \frac{1}{2}[1 - \text{sign}(x)]$.

- It may be also considered as an error correction process. This is the way it has been first put forward by Rosenblatt.

The perceptron as a categorizer

The set $\{I^{\mu, \text{in}}\}$ of input patterns is made of two subsets, $\{I^{\mu, \text{in}+}\}$ and $\{I^{\mu, \text{in}-}\}$:

$$\{I^{\mu, \text{in}}\} = \{I^{\mu, \text{in}+}\} \cup \{I^{\mu, \text{in}-}\}$$

such as the wanted response is

$$\xi^\mu = \begin{cases} +1 & \text{if } I^{\mu, \text{in}} \in \{I^{\mu, \text{in}+}\}, \\ -1 & \text{if } I^{\mu, \text{in}} \in \{I^{\mu, \text{in}-}\}. \end{cases}$$

In the limit of low noise levels the response of the unique output neuron is

$$\sigma^\mu = \text{sign}(h^\mu), \quad \text{with} \quad h^\mu = \sum_j J_j \xi_j^\mu.$$

We now present the first version of the *perceptron algorithm*.

- 1) If $\sigma^\mu = \xi^\mu$ then $\Delta J_j(\mu) = 0$.
- 2) If $\sigma^\mu = -\xi^\mu$ then
 - either $I^{\mu, \text{in}} \in \{I^{\mu, \text{in}+}\}$ and $\Delta J_j(\mu) = +\xi_j^\mu$
 - or $I^{\mu, \text{in}} \in \{I^{\mu, \text{in}-}\}$ and $\Delta J_j(\mu) = -\xi_j^\mu$.
- 3) Iterate in 1) with another pattern $I^{\mu'}$.

The perceptron algorithm.

We show that this learning dynamics is identical to that defined by Eq. (7.31):

The two rules 2) can be lumped:

$$2') \quad \text{If } \sigma^\mu = -\xi_j^\mu \text{ then } \Delta J_j(\mu) = \xi^\mu \xi_j^\mu,$$

and the rules 1) and 2') can be written in a single formula:

$$1') \quad \Delta J_j(\mu) = f(\mu) \xi^\mu \xi_j^\mu,$$

where

$$f(\mu) = \frac{1}{2}(1 - \xi^\mu \sigma^\mu) = \frac{1}{2} \left[1 - \text{sign} \left(\xi^\mu \sum_j J_j \xi_j^\mu \right) \right] = \frac{1}{2} (1 - \text{sign}(x^\mu)),$$

which proves the identity of definitions of perceptron rules.

The perceptron theorem

The theorem states that the perceptron rule always finds the set of connections J_j which correctly categorizes all the input patterns:

$$\sigma^\mu = \xi^\mu, \quad \forall \mu,$$

if such a solution exists.

Proof. — As in section 6.2.2, we define two vectors \tilde{J} and $\tilde{\tau}^\mu$ by

$$(\tilde{J})_j = J_j, \quad (\tilde{\tau}^\mu)_j = \xi^\mu \xi_j^\mu, \quad j = 0, 1, \dots, N, \quad \mu = 1, \dots, P.$$

The perceptron rule is written as:

$$\Delta \tilde{J}(\mu) = \frac{1}{2}(1 - \text{sign}(\tilde{J} \cdot \tilde{\tau}^\mu)) \tilde{\tau}^\mu = f(\mu) \tilde{\tau}^\mu, \quad (7.32)$$

with $f(\mu) \in \{0, 1\}$. If $\Delta \tilde{J}(\mu) \neq 0$, then $f(\mu) = 1$, but since

$$f(\mu) = \frac{1}{2}(1 - \text{sign}(\tilde{J} \cdot \tilde{\tau}^\mu)) = 1$$

one has $\tilde{J} \cdot \tilde{\tau}^\mu < 0$ and therefore $\tilde{J} \cdot \Delta \tilde{J}(\mu) < 0$. (7.33)

We now assume that a solution \tilde{J}^* exists. This amounts to saying that one can find sets of parameters J_{ij}^* , such that

$$\text{sign} \sum_j J_{ij}^* \xi_j^\mu = \xi_i^\mu.$$

This formula is written as

$$\tilde{J}^* \cdot \tilde{\tau}^\mu > 0, \quad \forall \mu. \quad (7.34)$$

It is also assumed without any lack of generality that the solution \tilde{J}^* is a normalized vector: $|\tilde{J}^*|^2 = N$ since if \tilde{J}^* is a solution $\lambda \tilde{J}^*$, ($\lambda > 0$) is also a solution. (This point is given more attention below.) Let $N\delta$ be the (positive) minimum of all quantities defined in Eq. (7.34):

$$N\delta = \min_\mu (\tilde{J}^* \cdot \tilde{\tau}^\mu).$$

δ is a measure of the largest angle between the solution vector (if it exists) and the various pattern vectors. The relation implies that

$$\Delta \tilde{J}(\mu) \cdot \tilde{J}^* \geq N\delta \quad (7.35)$$

since, if $\Delta\tilde{J}(\mu) \neq 0$ then $\Delta\tilde{J}(\mu) = \tilde{\tau}^\mu$. The angle between $\tilde{J}(\mu + 1)$ and \tilde{J}^* after the $(\mu + 1)$ th learning step is completed is given by

$$\cos(\theta(\mu + 1)) = \frac{\tilde{J}(\mu + 1) \cdot \tilde{J}^*}{|\tilde{J}(\mu + 1)| \cdot |\tilde{J}^*|}. \quad (7.36)$$

The numerator of this quantity is

$$\tilde{J}(\mu + 1) \cdot \tilde{J}^* = \tilde{J}(\mu) \cdot \tilde{J}^* + \Delta\tilde{J}(\mu) \cdot \tilde{J}^*,$$

or, with Eq. (7.35),

$$\tilde{J}(\mu + 1) \cdot \tilde{J}^* \geq \tilde{J}(\mu) \cdot \tilde{J}^* + N\delta > \mu N\delta.$$

Likewise,

$$\begin{aligned} |\tilde{J}(\mu + 1)|^2 &= |\tilde{J}(\mu) + \Delta\tilde{J}(\mu)|^2 \\ &= |\tilde{J}(\mu)|^2 + 2\tilde{J}(\mu) \cdot \Delta\tilde{J}(\mu) + |\Delta\tilde{J}(\mu)|^2. \end{aligned}$$

Using Eq. (7.33), it becomes

$$|\tilde{J}(\mu + 1)|^2 < |\tilde{J}(\mu)|^2 + |\Delta\tilde{J}(\mu)|^2 < |\tilde{J}(\mu)|^2 + N < \mu N.$$

Therefore

$$\cos(\theta(\nu + 1)) > \frac{N\mu\delta}{\sqrt{N\mu}\sqrt{N}} = \delta\sqrt{\mu},$$

which shows that the angle θ decreases as the number of learning steps, labeled by index μ , increases. But a cosine cannot be larger than 1 and therefore the number N_L of learning steps must be bounded. One has

$$N_L \leq \frac{1}{\delta^2}. \quad (7.37)$$

As long as the number P of patterns remains finite, $P \simeq O(N^O)$, the gap δ is finite and the number of steps which are necessary to make all patterns stable is finite whatever the size N of the network. On the other hand one knows, thanks to the Cover theorem, that at least one solution \tilde{J}^* always exists for $P < 2N$. The perceptron theorem makes certain that the algorithm finds one of the available solutions. The question remains of assessing how long it takes for the perceptron to find a solution in the limit $P \rightarrow 2N$. This amounts to finding δ in that limit. Calculations by Opper *et al.* indicate that $\delta \simeq 2 - \alpha$ with $\alpha = P/N$

and therefore the convergence time of the perceptron algorithm would scale as

$$\mathcal{N}_L \simeq (2 - \alpha)^{-2}.$$

7.4.3 Variations on the perceptron theme

The perceptron rule has been the source of a number of related algorithms and interesting developments which we review in this section.

a) The pocket algorithm

We associate a number of errors \mathcal{N}_E with a given set $\{J_j\}$ of interactions which connect the N^I input units j to the unique output unit. \mathcal{N}_E is the number of patterns I^μ such as $\sigma^\mu \neq \xi^\mu$:

$$\mathcal{N}_E = \sum_{\mu} \mathbf{1}(-x^\mu).$$

- 1) For a given set $\{J_j\}$ of interactions compute the number of errors \mathcal{N}_E .
- 2) Use the perceptron rule for a new pattern I^μ .
- 3) Use the new set of interactions $\{J_j(\mu)\}$ to compute the new number of errors $\mathcal{N}_E(\mu)$.
- 4) If $\mathcal{N}_E(\mu) < \mathcal{N}_E$ then $\{J_j\} = \{J_j(\mu)\}$ and $\mathcal{N}_E = \mathcal{N}_E(\mu)$ and go to step 2)
else go to step 2).

The pocket algorithm.

The perceptron rule is a serial learning dynamics which strives to stabilize the last pattern I^μ which the system experiences. If this step is successful x^μ becomes positive. One could conclude that \mathcal{N}_E decreases ($\mathcal{N}_E \mapsto \mathcal{N}_E - 1$), but the stabilization of I^μ can bring about the destabilization of formerly stable patterns even though, as proved by the perceptron theorem, the process finally converges to the stabilization of all patterns (if a solution \tilde{J}^* exists). The pocket algorithm consists in giving up all perceptron steps whose results are increases of \mathcal{N}_E : as learning proceeds, one keeps in one’s pocket the set $\{J_j\}$ of interactions which have generated the smallest number of errors \mathcal{N}_E so far.

The number of errors \mathcal{N}_E is a positive, non-increasing quantity. One is therefore certain that the process must stop. Since the perceptron theorem is still applicable to this learning rule, the pocket algorithm finds a solution \tilde{J}^* if such a solution exists. If a solution does not exist the algorithm yields a set of interactions which is ‘as good as possible’

in the sense that it is the set which makes the number of ill-categorized patterns close to its minimum value.

b) *The minover algorithm*

The idea of the minover algorithm, which is due to Krauth and Mézard, is to apply the perceptron rule to the less stable pattern. A given a set of interactions determines the set $\{x^\mu\}$ of stabilization parameters. Among the patterns to be categorized we choose the pattern I^{μ_0} such that

$$x^{\mu_0} < x^\mu, \quad \forall \mu \neq \mu_0,$$

and the perceptron rule is applied to I^{μ_0} .

- 1) For a given set $\{J_j\}$ of interactions compute the P stabilization parameters $\{x^\mu\}$.
- 2) Find the pattern I^{μ_0} such that $x^{\mu_0} < x^\mu$ for all $\mu \neq \mu_0$.
- 3) Apply the perceptron rule to pattern I^{μ_0} .
- 4) Iterate in 1).

The minover algorithm.

The perceptron theorem applies to the minover algorithm: its convergence is secured. When compared with the classical perceptron algorithm, the minover algorithm runs faster because learning is always applied to the worst situation (that is to say, to the most unstable pattern).

c) *Application to fully connected networks*

Wallace and Gardner have realized that the application of the perceptron algorithm is not limited to the simple architecture we have considered so far. For example, a fully connected network may be viewed as a set of N independent perceptrons. Each perceptron, labeled perceptron i , is made of one output unit i , which is one of the N units of the network, and N input units, which are the $(N - 1)$ remaining units supplemented by a threshold unit. In this system the connection J_{ij} is independent of the connection J_{ji} . When the system does not comprise hidden neurons, a state ξ_i^μ of pattern I^μ is both the output state for perceptron i and an input state for all the other perceptrons. If a given pattern I^μ is stable for all perceptrons it is stable for the whole system, which means that I^μ is a fixed point of the neural dynamics (a memorized state).

The Cover theorem guarantees that for each perceptron, there exists a solution \tilde{J}_i^* which stabilizes P random patterns provided $P < 2N$. If a solution exists the perceptron theorem makes certain that the N

perceptrons find a solution, that is to say a connection matrix \mathbf{J} which stabilizes the P patterns. The conclusion is that the maximum memory storage capacity that the use of the perceptron algorithm is able to achieve is

$$\alpha_c = \frac{P_c}{N} = 2,$$

and that this is a limit that cannot be trespassed for random patterns.

The perceptron rule generates asymmetrical interactions. We see here an example of a system whose dynamics has a number of fixed points as limit behaviors and yet is asymmetrically connected.

The perceptron rule can be symmetrized

$$\Delta J_{ij}(\mu) = f_{ij}(\mu) \xi_i^\mu \xi_j^\mu,$$

with $f_{ij}(\mu) = f_i(\mu) + f_j(\mu) = 1 - \frac{1}{2} [\text{sign}(x_i^\mu) + \text{sign}(x_j^\mu)]$, but this is not a good idea. The perceptron rule strives to stabilize the patterns at best and any constraint such as compelling the interactions to be symmetrical $J_{ij} = J_{ji}$ damages the stability as shown by numerical simulations.

d) The perceptron rule and weakly constrained interactions

Equation (7.32) suggests a geometrical interpretation of the perceptron algorithm. We consider an N -dimensional space that is viewed as the support of $\tilde{\mathbf{J}}$, the current vector of synaptic efficacies $J_j = (\tilde{\mathbf{J}})_j$, impinging on the output unit, that of the set of P vector $\tilde{\tau}^\mu$, one for each memorized pattern I^μ and that of the P μ -planes defined by

$$\tilde{\mathbf{J}} \cdot \tilde{\tau}^\mu = 0.$$

A pattern I^μ is stable when the inequality

$$x^\mu = \tilde{\mathbf{J}} \cdot \tilde{\tau}^\mu > 0 \quad (7.38)$$

is satisfied, that is to say when the vector $\tilde{\mathbf{J}}$ lies on the good side of the corresponding μ -plane. If the case arises $\Delta \tilde{\mathbf{J}}(\mu) = 0$ and the vector $\tilde{\mathbf{J}}$ does not move. Otherwise the rule pushes $\tilde{\mathbf{J}}$ towards the corresponding μ -plane along a direction which is parallel to that of $\tilde{\tau}^\mu$, that is to say a direction which is perpendicular to the μ -plane (see Fig. 7.7).

If the rule succeeds in stabilizing all the patterns, that is if

$$x^\mu = \tilde{\mathbf{J}} \cdot \tilde{\tau}^\mu > 0, \quad \forall \mu,$$

the vector $\tilde{\mathbf{J}}$ finds itself in the polyhedral convex cone determined by the P inequalities (7.38). We call this cone a V -cone (V for Venkatesh, who was the first to apply the Cover theorem to neural networks). The perceptron theorem states that $\tilde{\mathbf{J}}$ gets closer and closer to the V -cone

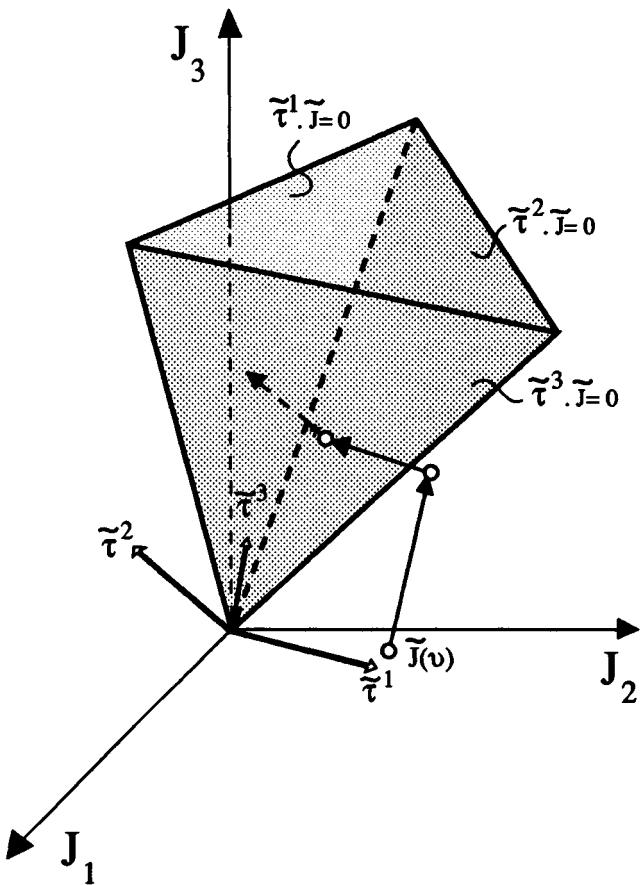


Figure 7.7. The schematic trajectory of \tilde{J} in the space of interactions associated with i . It is a piecewise trajectory whose pieces are vectors $\tilde{\tau}^\mu$. It stops once it enters the V -cone.

until it finally penetrates into the cone. Once in the cone \tilde{J} does not move any more.

It may appear interesting to normalize the length of vector \tilde{J} , that is to say to make it obey the weak constraint (7.2) (see section 7.2.1),

$$\sum_j |J_j|^2 = |\tilde{J}|^2 = L,$$

since the stability conditions (7.38) are not changed by a mere renormalization of the length of \tilde{J} . The trajectory of \tilde{J} is now compelled to remain on a sphere of radius \sqrt{L} . It is a piecewise curve which stops when entering the spherical polyhedron defined by the intersection of the V -cone with the sphere (see Fig. 7.8). This normalization has been used in the calculation of the volumes of solutions in section 6.3. We now consider the problem of the dynamics of synaptic efficacies driven by a constrained perceptron rule.

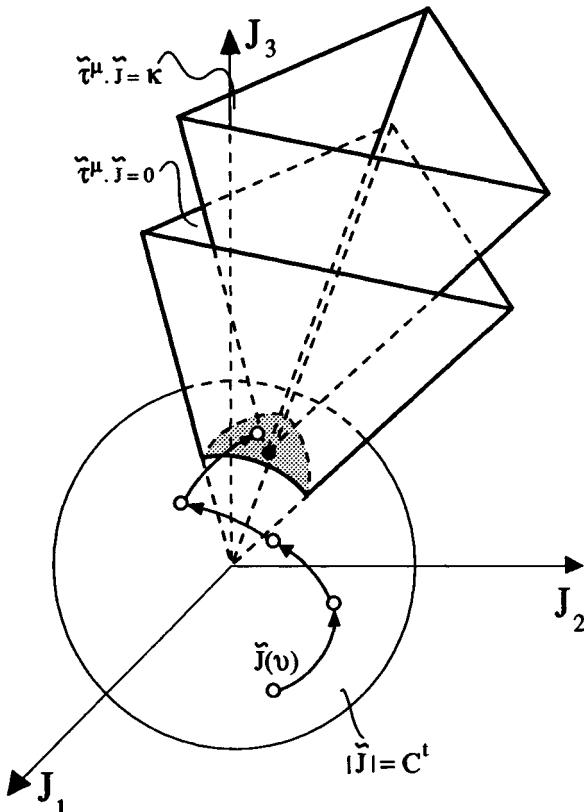


Figure 7.8. Introducing a bias κ in the perceptron rule allows a better stabilization of memorized patterns.

We first define normalized stabilization parameters

$$x^\mu = \frac{\sum_j J_j \xi^\mu \xi_j^\mu}{\sqrt{\sum_j (J_j)^2}},$$

and we use these normalized parameters in the definition of the cost

function $H(I^\mu)$:

$$H(I^\mu) = g(x^\mu) \quad \text{with} \quad g(x) = \begin{cases} -x & \text{if } x < 0, \\ 0 & \text{if } x > 0. \end{cases}$$

The gradient dynamics gives

$$\Delta J_j(\mu) = \varepsilon f(\mu) \left(\frac{\xi^\mu \xi_j^\mu}{\sqrt{\sum_k (J_k)^2}} - \frac{x^\mu J_j}{\sum_k (J_k)^2} \right),$$

where $f(\mu) = \mathbf{1}(-x^\mu)$. Let us compute the quantity

$$\begin{aligned} \sum_j J_j \Delta J_j &= \varepsilon f(\mu) \left[\frac{\sum_j J_j \xi^\mu \xi_j^\mu}{\left(\sum_k (J_k)^2 \right)^{1/2}} - \frac{x^\mu \sum_j (J_j)^2}{\sum_k (J_k)^2} \right] \\ &= \varepsilon f(\mu) (x^\mu - x^\mu) = 0, \end{aligned}$$

which proves that the gradient dynamics respects the weak constraint. Finally the learning rule is given by

$$\Delta J_j(\mu) = \varepsilon' \mathbf{1}(-x^\mu) \left(\xi^\mu \xi_j^\mu - \frac{1}{T^\mu} J_j \right),$$

with $\varepsilon' = \varepsilon/\sqrt{L}$ and $(T^\mu)^{-1} = x^\mu/\sqrt{L}$.

We observe that the constraint manifests itself by the introduction of a relaxation term in the learning dynamics.

e) Biased perceptron rule

The problem with the perceptron rule is that the synaptic efficacies stop changing once the vector \tilde{J} crosses the μ -planes and the radii of basins of attraction are essentially zero. To improve the algorithm it is necessary to push the vector \tilde{J} further inside the V -cone. This can be carried out by introducing a bias $\kappa > 0$ (not to be confused with the thresholds θ_i) in the stability conditions (7.38):

$$x^\mu = \tilde{J} \cdot \tilde{\tau}^\mu > \kappa. \quad (7.39)$$

This is another idea which we owe to E. Gardner and which has already been introduced in the computation of the volumes of solutions in the phase space of interactions (section 6.3). The vector \tilde{J} now stops moving when it is inside a cone determined by planes parallel to those of the V -cone and at a distance κ from these planes (see Fig. 7.7). In the cost function the stabilization parameters have simply to be replaced by $x^\mu - \kappa$. Since the motion of \tilde{J} is, moreover, constrained by the weak condition, the dynamics becomes

$$\Delta J_j(\mu) = \varepsilon \mathbf{1}(\kappa - x^\mu) \left(\xi^\mu \xi_j^\mu - \frac{1}{T^\mu} J_j \right),$$

with $(T^\mu)^{-1} = x^\mu / \sqrt{L}$. As the bias κ increases the convex polyhedron determined by the intersection of the translated V -cone and of the normalization sphere shrinks. The best bias κ is the bias which corresponds to a V -cone with its summit right on the sphere (see Fig. 7.7).

f) Continuous learning dynamics

For the perceptron learning principle to be satisfied it is necessary that the function $g(x)$ which determines the cost function H ,

$$H = \sum_\mu g(x^\mu), \quad (7.40)$$

belongs to a family \mathcal{G} of functions characterized by two following properties:

- 1) $g(x)$ are monotonous decreasing functions.
- 2) $g(x \rightarrow +\infty) = C^t$.

This is equivalent to saying that $f(x) = dg(x)/dx$ belongs to a family \mathcal{F} of functions characterized by

- 1) $f(x) < 0$,
- 2) $f(x \rightarrow +\infty) = 0$.

The first property stems from the idea that the stabilization parameter

$$x^\mu = \xi^\mu h(I^\mu) = \sum_j J_j \xi^\mu \xi_j^\mu \quad (7.41)$$

is a measure of the depth of the basin of I^μ at output site; the larger the parameter, the deeper the basin. If one assumes that the size of the basin is a monotone increasing function of its depth x^μ , maximizing x^μ will amount to maximizing the sizes of the basins. This assumption is supported by numerical simulations carried out by Forrest on pure Hebbian models.

The second property is imposed by the perceptron principle which forbids the network to learn an already well imprinted pattern: a pattern whose basin of attraction is large does not need further stabilization.

The gradient dynamics yields

$$\frac{d\tilde{J}}{dt} = - \sum_\mu \tilde{\tau}^\mu f(\tilde{J} \cdot \tilde{\tau}^\mu). \quad (7.42)$$

Learning is now a parallel and continuous process: all patterns are learned together at every time step.

One recovers the various forms of learning rules by appropriate choices of the function $g(x)$.

1) We have already seen that if $g(x) = -x$ the Hopfield rule is recovered. One finds

$$\tilde{J} = \sum_{\mu} \tilde{\tau}^{\mu},$$

which means that the (Hopfield) efficacies vector \tilde{J} of synapses impinging on a given neuron is the vector sum of the various pattern vectors $\tilde{\tau}^{\mu}$. This choice is not convenient, since the function $f(x) = -1$ does not follow the second property of \mathcal{F} .

2) Letting $g(x) = x(x - 2)$ leads to the Widrow Hoff rule. With this choice we saw that the steady solutions are given by

$$\tilde{J} \cdot \tilde{\tau}^{\mu} = 1, \quad \forall \mu.$$

\tilde{J} is therefore the P -sector of the polyhedron determined by the P patterns. The definition of a P -sector is meaningful only for independent vectors and therefore this solution holds only when $P < N$. However, the function $f(x) = 2(x - 1)$ is at odds with the two properties of \mathcal{F} and therefore it cannot yield the optimal solution.

3) Choosing $g(x) = -x$ for $x < 0$ and $g(x) = 0$ for $x > 0$ is a better choice but the function is flat for positive x 's, which brings the dynamics to a halt before achieving optimal solutions: as soon as the synaptic vector \tilde{J} enters the V -cone it stops. This is the Wallace Gardner algorithm with zero bias. Introducing a bias κ amounts to choosing $g(x) = -x$ ($f(x) = 1$) for $x < \kappa$ and $g(x) = -\kappa$ ($f(x) = 0$) for $x > \kappa$.

4) An interesting choice for g is (Peretto)

$$g(x) = \exp(-\lambda x).$$

The dynamics of \tilde{J} is driven by a set of forces $f(x) = \lambda \exp(-\lambda x)$ which push away the extreme end of \tilde{J} from the various μ -planes; the closer the plane the larger the expelling force. This model shows therefore some similarity to the minover algorithm.

Up to now we have paid no attention to the normalization of synaptic vectors. For the efficacies to obey (strict) weak constraints it is enough to add a relaxation term to the equation of the synaptic dynamics (7.42):

$$\Delta J_j = \varepsilon \sum_{\mu} \exp(-\lambda x^{\mu}) \left(\xi^{\mu} \xi_j^{\mu} - \frac{J_j}{T^{\mu}} \right),$$

but it is more interesting to assume that the weak constraints are loose:

$$\sum_j (J_j)^2 \leq L. \quad (7.43)$$

The norm of synaptic efficacies is no more fixed. It remains free to vary as long as the ‘crystallization sphere’ given by Eq. (7.43) is not reached. If the case arises, that is to say if at some moment the norm $|\tilde{J}|^2 = L$, then it stops moving permanently. This process displays some interesting properties which can be used to build a model of short- and long-term memories. In particular it provides a mean of automatically avoiding the catastrophe that results from overcrowding.

Let P patterns I^μ be given.

- If the corresponding V -cone does not exist, which implies that $P > 2 \times N$ for random patterns, all the fixed points of the synaptic gradient dynamics lie at finite range ($|\tilde{J}|$ remains finite) even in an unconstrained system ($L \rightarrow \infty$).
- If a V -cone does exist, the asymptotic behavior of the synaptic dynamics comprises at least one fixed point at infinite range inside the V -cone. In this case the trajectory stops on the hypersphere associated with the weak constraint Eq. (7.43).

Proof. — The proof is geometrical in nature. Let us consider the evolution of the length of \tilde{J} . Using the Eq. (7.13), the dynamics of the length obeys

$$\frac{1}{2} \frac{d|\tilde{J}|^2}{dt} = -\varepsilon \sum_\mu \tilde{J} \cdot \tilde{\tau}^\mu f(\tilde{J} \cdot \tilde{\tau}^\mu). \quad (7.44)$$

If there is no V -cone, at least one of the projections $\tilde{J} \cdot \tilde{\tau}^\mu$ is negative. Therefore there exists a non-empty set of large components in the r.h.s. of Eq. (7.44). Let us study the trajectories originating from the points of a sphere of radius r . For finite sizes N of the network it is always possible to find a large but finite r such that the positive components overcome all the other components of Eq. (7.44). Therefore there always exists a sphere of large enough radius r such that the trajectory of \tilde{J} flows inwards on every point of its surface. This proves the first statement.

Let us assume now that a V -cone does exist. The argument we saw above still holds for the points of the sphere which are inside the Venkatesh cone, with the conclusion now that the trajectories flow out from the sphere (in fact more and more slowly as r increases). Let us choose a point on the cone, for instance on the plane defined by

$$\tilde{J} \cdot \tilde{\tau}^\mu = 0.$$

The dynamics of the component of \tilde{J} along $\tilde{\tau}^1$ is given by

$$\frac{dx^1}{dt} = \frac{d(\tilde{J} \cdot \tilde{\tau}^1)}{dt} = -\varepsilon \left[N f(0) + \sum_{\mu>1} \tilde{\tau}^1 \cdot \tilde{\tau}^\mu f(\tilde{J} \cdot \tilde{\tau}^\mu) \right]. \quad (7.45)$$

All projections $\tilde{J} \cdot \tilde{\tau}^\mu$ entering Eq. (7.45) are positive. Therefore for large enough r the first term dominates, and the trajectories flow in towards the cone. This proves that when a V -cone exists there is always at least one fixed point at infinite range inside the cone (Fig. 7.9).

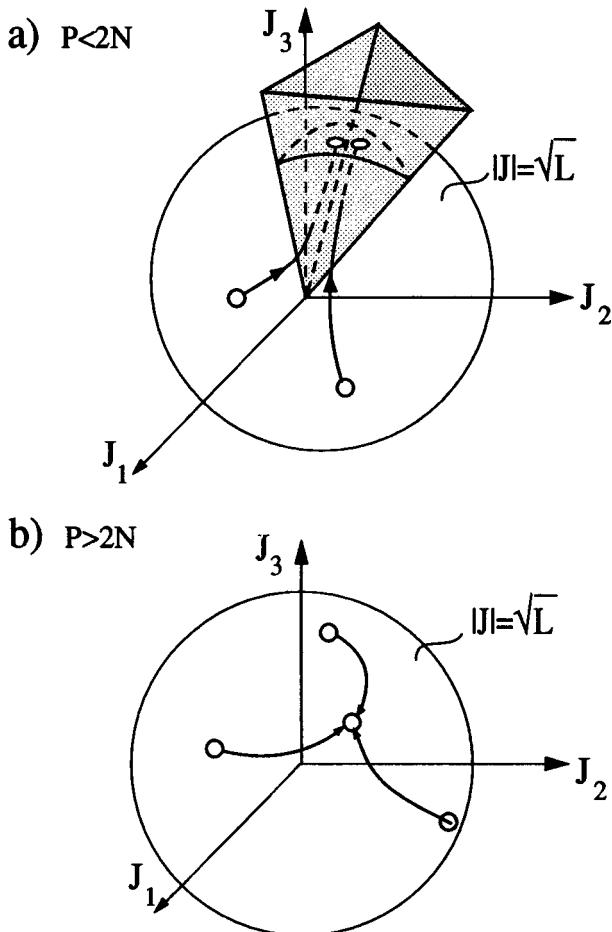


Figure 7.9. Typical trajectories of the synaptic vector J in the phase space of interactions.

- a) $P < 2 \times N$: there exist centrifugal trajectories lying inside the V -cone. They eventually stop on the crystallization sphere.
- b) $P > 2 \times N$: the trajectories flow to fixed points lying at finite ranges. They cannot reach the crystallization sphere.

Let us now assume that the network is to learn two sets of random patterns.

The first set \mathcal{E}^1 comprises $N^1 > 2 \times N$ patterns and the second set \mathcal{E}^2 comprises $N^2 < 2 \times N$ patterns. The system is unable to store the first set \mathcal{E}^1 since it is overcrowded, but the dynamics keeps the vector $\tilde{\mathbf{j}}$ far from the crystallization sphere and the efficacies do not get frozen during this learning stage. The crystallization (long-term memory) process may only occur for sets such as \mathcal{E}^2 which are learnable even though the system has experienced unlearnable sets before.

This behavior is in contrast with pure Hebbian algorithms whose efficacies may freeze for unlearnable training sets or with strictly constrained networks which cannot be endowed with long-term memory properties.

The computer simulations confirm those results: the change in the nature of asymptotic behaviors is observed for $P = 2 \times N$ (see Fig. 7.10). They also show that the dynamics in effect allows $2 \times N$ patterns to be stored. Finally the dynamics succeeds in building large basins of attraction, the goal for which it has been devised (see Fig. 7.11).

7.5 Correlated patterns

So far we have been concerned with the learning of uncorrelated patterns. Patterns I^μ are uncorrelated if the distribution of the elements $\Gamma_{\mu\mu'}$ of the correlation matrix Γ is Gaussian with *zero mean value*. But let us imagine that the patterns are characters drawn on a piece of paper. The characters I^μ are assumed to be centered and determined by pixels whose states are $\xi_i^\mu = -1$. The states of pixels which depict the background are $\xi_i^\mu = +1$. Since the overlaps between the backgrounds of the various characters are so large, the patterns are far from being uncorrelated. This is a situation which is very often encountered in associative learning and the problem of correlation is therefore central. It is given some attention in this chapter.

7.5.1 Learning correlated patterns with the pure Hebbian rule

Let m be the average overlap between the characters and a uniform background (a configuration with all units i in state $+1$). For the moment one assumes that there does not exist any other correlation between the patterns and therefore that the distribution of states is given by

$$P(\xi_i^\mu) = \frac{1}{2}(1+m)\delta(\xi_i^\mu - 1) + \frac{1}{2}(1-m)\delta(\xi_i^\mu + 1), \quad (7.46)$$

with $-1 < m < +1$.

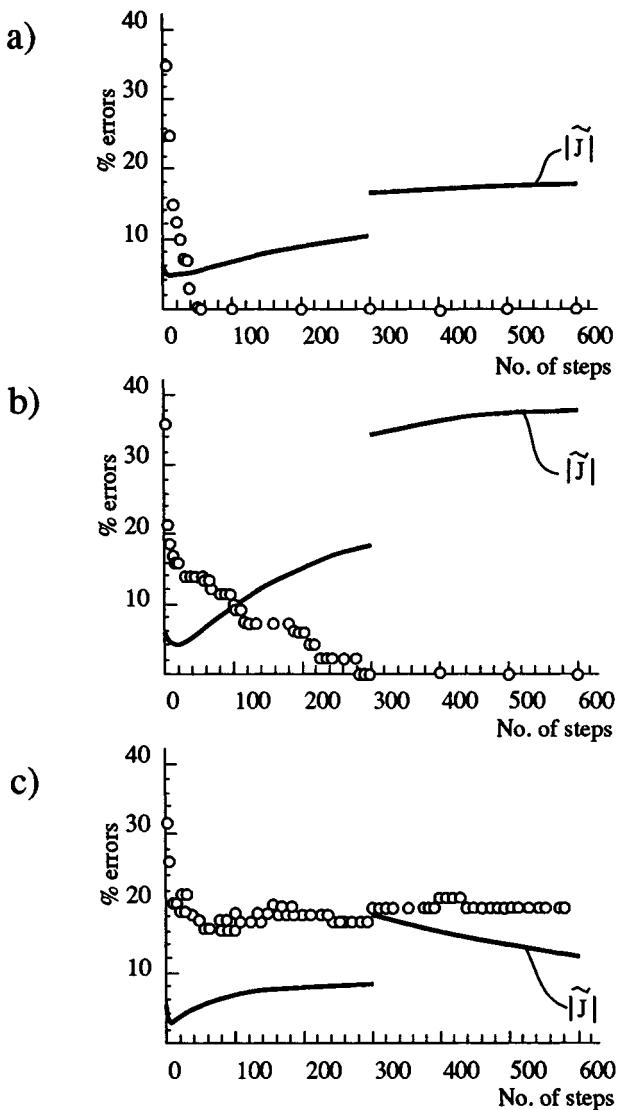


Figure 7.10. Three simulations.

- For $P/N < 1$ the learning algorithm converges fast and the module of the synaptic vector keeps increasing.
- Slightly below the Cover limit for $P/N = 7/4$, the convergence is much slower. The module still increases.
- For $P/N = 9/4$, above the Cover limit ($P/N = 2$), no convergence is achieved. The module of $|\tilde{J}|$ is bounded. This is made conspicuous by doubling $|\tilde{J}|$ after a number of steps.

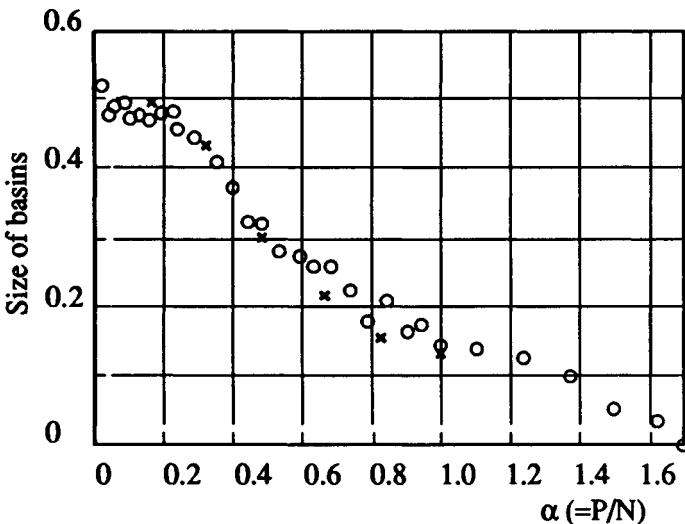


Figure 7.11. Average sizes of basins of attraction as a function of the number of memorized patterns.
 $\circ : N = 40$; $\times : N = 60$. The size is defined as the average number of wrong bits which allows the trajectory to escape the basin.

Remarks

a) The choice of a background state $I^{\text{back}} = \{\xi_i^{\text{back}} = +1\}$ (I^{back} does not belong to the set \mathcal{E} of memorized states) as the reference state may seem special, but a mere gauge transformation $\sigma_i \mapsto \xi_i^{\text{ref}} \sigma_i$ changes any reference state $I^{\text{ref}} = \{\xi_i^{\text{ref}}\}$ into the uniform state I^{back} .

b) One of the most serious criticisms that biologists address to the theory of neural networks we have developed so far is that the level of activities of the model neurons is, on average, half of their maximum firing rates; that is, the frequencies of model neurons are about a hundred spikes per second. The observed frequencies, however, are smaller by an order of magnitude. One way of reducing the overall activity of the system is to assume that the stored patterns are comprised of many more -1 values (silent neurons) than $+1$ values (firing neurons). Here the background is to be taken as $I^{\text{back}} = \{\xi_i^{\mu} = -1\}$. In reality this way of tackling the problem of low activities is not very convenient since, if most neurons are actually silent, the remaining units are still very active, which is not what is observed. There are other models to deal with low activities which we will consider in the last chapter.

The average number of pixels $\xi_i^{\mu} = -1$ in a character I^{μ} is

$$\frac{1}{2}(1-m)N.$$

The average activity per neuron in a memorized state is

$$\bar{\xi}_i^\mu = m$$

(if all neurons are silent then $m = -1$) and the overlap between two patterns is given by

$$I^\mu \cdot I^{\mu'} = \sum_{i=1}^N \xi_i^\mu \xi_i^{\mu'} = N m^2, \quad \mu \neq \mu'.$$

The correlated patterns can be learned by appealing to the algorithms which we have seen in the preceding sections, and the question arises as to how the correlations change the memory storage capacities.

- If one simply uses the Hebbian rule the storage capacity shrinks dramatically. The field on the neuron i when the system is in state I^1 is given by

$$h_i(I^1) = \frac{1}{N} \sum_j \sum_\mu \xi_i^\mu \xi_j^\mu \xi_j^1 = \frac{1}{N} \xi_i^1 \sum_j 1 + \frac{1}{N} \sum_{\mu \neq 1} \xi_i^\mu \xi_j^\mu \xi_j^1,$$

whence

$$h_i(I^1) = \xi_i^1 + \frac{1}{N} \sum_{\mu \neq 1} (I^\mu \cdot I^1) \xi_i^\mu, \quad (7.47)$$

with $I^\mu \cdot I^1 = N m^2$ and $\frac{1}{N} \sum_\mu \xi_i^\mu = \frac{Pm}{N}$ and therefore

$$h_i(I^1) = \xi_i^1 + m^3 P.$$

Since $m > 0$, the state of all units becomes $\sigma_i = +1$ as soon as the number of patterns becomes larger than P_c , a *finite number of patterns* with $P_c \simeq 1/|m|^3$.

- This catastrophic situation does not arise when projection algorithms are used. The very essence of the projection algorithms is to avoid correlations by making all patterns orthogonal. The field on site i is correlation-independent,

$$h_i(I^1) = \xi_i^1,$$

and the memory capacity is $\alpha_c = P_c/N = 1$, whatever m . The effect of correlations is to decrease the size of basins of attraction in a way that we shall not discuss here.

7.5.2 Associative learning rules for correlated patterns

The problem introduced by the existence of correlations is due to the fact that the noise term in Eq. (7.47) does not average zero. Amit *et al.*

proposed a modified Hebbian learning rule to correct this defect. This rule is

$$J_{ij} = \frac{1}{N} \sum_{\mu} (\xi_i^{\mu} - m)(\xi_j^{\mu} - m), \quad (7.48)$$

with $\xi_i^{\mu} \in \{+1, -1\}$. The meaning of m is the same as in the last section, that is to say m is a common overlap between the patterns and a special background pattern. According to Eq. (7.48) the interactions remain symmetrical, which leaves the model amenable to the techniques of statistical physics. Here we shall be content with a crude estimate of memory capacities of the network. When the network is in state I^1 , the local field is now given by

$$h_i(I^1) = \frac{1}{N} \sum_j (\xi_i^1 - m)(\xi_j^1 - m)\xi_j^1 + \frac{1}{N} \sum_j \sum_{\mu \neq 1} (\xi_i^{\mu} - m)(\xi_j^{\mu} - m)\xi_j^1.$$

The first term is the signal $h_i^s = (\xi_i^1 - m) \frac{1}{N} \sum_j (1 - m \xi_j^1)$. Its projection along the component of the first pattern is

$$h_i^s \xi_i^1 = (1 - m \xi_i^1) \frac{1}{N} \sum_j (1 - m \xi_j^1) = (1 - m \xi_i^1) \left[1 - \frac{m}{N} \sum_j \xi_j^1 \right],$$

and therefore $h_i^s \xi_i^1 = (1 - m \xi_i^1)(1 - m^2)$.

One observes that the coherent field increases on sites i where ξ_i^1 is opposite to m . The first and the second moment of the noise term h_i^n are derived from Table 7.1, where the quantity $X(C)$ is defined by

$$X(C) = (\xi_i^{\mu} - m)(\xi_j^{\mu} - m)\xi_j^1.$$

C is one of the eight possible configurations $\{\xi_i^{\mu}, \xi_j^{\mu}, \xi_j^1\}$. $P(C)$ is the probability of configuration C occurring.

We compute the first moment of the noise term:

$$\overline{h_i^n} = \frac{P}{N} \sum_C X(C) P(C) = 0.$$

It vanishes as it has to. The second moment is given by

$$\overline{(h_i^n)^2} = \frac{P}{N} \sum_C X^2(C) P(C),$$

whence $\overline{(h_i^n)^2} = \frac{P}{N}(1 - m^2)^2$.

	$\xi_i^\mu \xi_j^\mu \xi_j^1$	$X(C)$	$X^2(C)$	$P(C)$
C	+++	$(1 - m)^2$	$(1 - m)^4$	$\frac{1}{8}(1 + m)^3$
	++-	$-(1 - m)^2$	$(1 - m)^4$	$\frac{1}{8}(1 + m)^2(1 - m)$
	+--	$-(1 - m^2)$	$(1 - m)^2(1 + m)^2$	$\frac{1}{8}(1 + m)^2(1 - m)$
	-++	$-(1 - m^2)$	$(1 - m)^2(1 + m)^2$	$\frac{1}{8}(1 + m)^2(1 - m)$
	+--	$(1 - m^2)$	$(1 - m)^2(1 + m)^2$	$\frac{1}{8}(1 + m)(1 - m)^2$
	-+-	$(1 - m^2)$	$(1 - m)^2(1 + m)^2$	$\frac{1}{8}(1 + m)(1 - m)^2$
	--+	$(1 + m)^2$	$(1 + m)^4$	$\frac{1}{8}(1 + m)(1 - m)^2$
	--	$-(1 + m)^2$	$(1 + m)^4$	$\frac{1}{8}(1 - m)^3$

Table 7.1.

The largest number of patterns that the system is able to store is reached when the noise amplitude matches the signal amplitude. This occurs when

$$(h_i^s)^2 = (1 - |m|)^2 (1 - m^2)^2 = \overline{(h_i^n)^2} = \frac{P}{N} (1 - m^2)^2$$

and therefore the memory storage capacity is given by

$$P_c = N (1 - |m|)^2. \quad (7.49)$$

The capacity is proportional to the size of the network, but it decreases sharply when m increases. This situation can greatly be improved, as we shall see in section 7.5.4.

7.5.3 Storing hierarchically correlated patterns

Up to now the patterns to be memorized have been given no structures except for a bias m . But patterns have structures: different patterns share common features and a pattern is characterized by a given set of features. In many instances data are ordered according to tree-like organizations with every level of the tree associated with a feature. Let us assume, for example, that there exist three sorts of features, μ , ν and η . A state $I^{\eta(\nu(\mu))}$ is a state which is a member of a large class of systems whose common feature is μ . It also belongs to a sub-class of this class which is characterized by the feature $\nu(\mu)$. Finally, the last label $\eta(\nu(\mu))$ points out the system itself. This state can be thought of as being a leaf of a tree comprising three levels, characterized by the

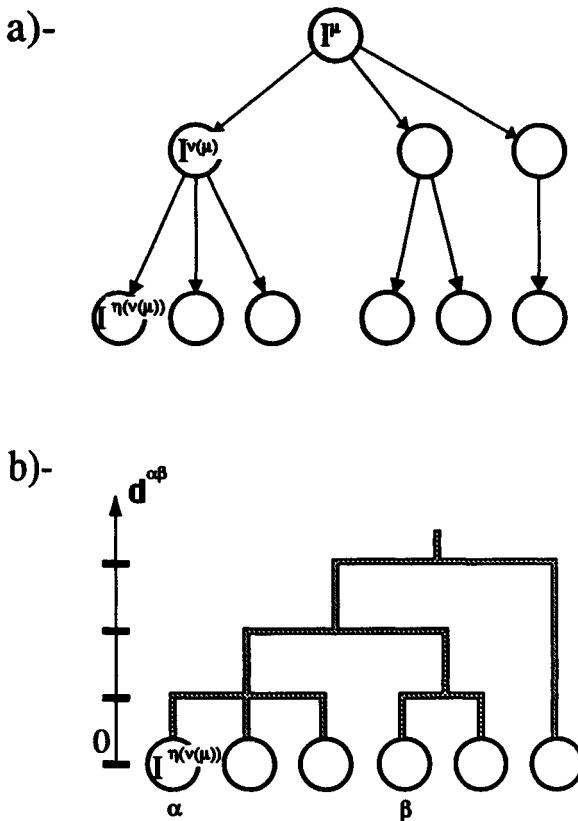


Figure 7.12. A family tree of features and ultrametric distances.

labels μ , ν and η respectively (see Fig. 7.12). Hierarchical structures are ubiquitous in many areas of human activities. To give a few examples, data processing uses techniques which mainly aim at finding hierarchical organizations hidden in raw data, and artificial intelligence approaches to pattern recognition are essentially based on hierarchically structured searches. On the other hand, we have seen in Chapter 2 that the cortex is somehow hierarchically organized, with small groups of neurons making microcolumns, microcolumns gathered in columns, columns in turn making maps and finally maps organized in cortical areas. This suggests that knowledge may also be hierarchically imprinted in the cortex, but at present nobody knows if the case ever arises.

Hierarchical trees are similar to family trees. In these structures one defines the distance between two leaves (two states) as the number of

layers leading to the closest ancestor. Given three states I^μ , $I^{\mu'}$ and $I^{\mu''}$, the three distances $d^{\mu\mu'}$, $d^{\mu'\mu''}$ and $d^{\mu''\mu}$ between the states must obey the inequality,

$$d^{\mu\mu'} \leq \max(d^{\mu'\mu''}, d^{\mu''\mu}),$$

as well as the two other inequalities which one obtains through permutation of indices. For the three inequalities to be simultaneously satisfied it is necessary that two out of the three distances are equal, and larger than or equal to the third one. A space with distances obeying such a rule is called an *ultrametric space*.

In neural networks the states I are defined by strings of N binary variables and the distance $d^{\mu\mu'}$ between two states I^μ and $I^{\mu'}$ is the *Hamming distance*, the number of individual neuronal states which are different in I^μ and $I^{\mu'}$ (see Eq. 7.24).

The problem is first to build ultrametric neuronal states and then to find a rule which allows an efficient storage of these patterns.

Ultrametric states are built by processes in which a parent gives birth to various offspring. This can be done in two ways:

a) *Process P1.* — By adding new individual states to old ones. We start from a number of random states (the first generation) comprising N_1 neurons each. Then we consider a large number of random states comprising N_2 neurons. A state of the second generation, comprising $N_1 + N_2$ states, is obtained by a concatenation of one given state of the first generation with one state of the second generation. The process is repeated to generate a whole family of ultrametric states comprising $N = N_1 + N_2 + \dots$ neurons. The Hamming distance between two ‘brothers’ is $\frac{1}{2}N_2$, whereas the distance between two ‘first cousins’ is $(N_1 + \frac{1}{2}N_2)$.

b) *Process P2.* — By modifying the states at every generation. This process has been put forward by Gutfreund. We start from states I^μ of N neurons which are generated according to the distribution (7.46):

$$P(\xi_i^\mu) = \frac{1}{2}(1 + m^0)\delta(\xi_i^\mu - 1) + \frac{1}{2}(1 - m^0)\delta(\xi_i^\mu + 1), \quad m^0 > 0.$$

These are the states of the first generation. Let us choose one of these states, $I^\mu = \{\xi_i^\mu\}$. A state of N neurons of the second generation is obtained by using the following probability distribution:

$$\begin{aligned} P(\xi_i^{\mu\nu}) = & \frac{1}{2}(1 + \xi_i^\mu m^1)\delta(\xi_i^{\mu\nu} - 1) \\ & + \frac{1}{2}(1 - \xi_i^\mu m^1)\delta(\xi_i^{\mu\nu} + 1), \quad m^1 > 0, \end{aligned} \tag{7.50}$$

with $\xi_i^{\mu\nu} \in \{+1, -1\}$, which implies that, in the second generation, $\xi_i^{\mu\nu}$ is more likely to align along ξ_i^μ than along the opposite direction $-\xi_i^\mu$.

According to this process one has

$$I^{\mu\nu} \cdot I^{\mu'\nu'} = \begin{cases} N(m^1)^2 & \text{if } \mu = \mu', \nu \neq \nu', \\ N(m^0)^2(m^1)^2 & \text{if } \mu \neq \mu', \end{cases}$$

which shows the ultrametric character of the states. Also,

$$I^{\mu\nu} \cdot I^\mu = Nm^1 \quad \text{and} \quad I^{\mu\nu} \cdot I^{\mu'} = N(m^0)^2(m^1) \quad \text{if} \quad \mu \neq \mu'.$$

The patterns are stored according the following learning rule (Gutfreund):

$$\begin{aligned} J_{ij} &= \frac{1}{N} \sum_{\mu,\nu} (\xi_i^{\mu\nu} - m^1 \xi_i^\mu)(\xi_j^{\mu\nu} - m^1 \xi_j^\mu), \\ \mu &= 1, \dots, P_0, \quad \nu = 1, \dots, P_1. \end{aligned}$$

The total number of patterns is $P = P_0 P_1$. Here only hierarchies with two levels are considered. The extension to more complicated hierarchies is straightforward, but it induces lengthy strings of indices. When the system is in state I^{11} the local field is given by

$$\begin{aligned} h_i(I^{11}) &= \frac{1}{N} \sum_j (\xi_i^{11} - m^1 \xi_i^1)(\xi_j^{11} - m^1 \xi_j^1) \xi_j^{11} \\ &\quad + \frac{1}{N} \sum_j \sum_{\mu\nu \neq 11} (\xi_i^{\mu\nu} - m^1 \xi_i^\mu)(\xi_j^{\mu\nu} - m^1 \xi_j^\mu) \xi_j^{11}. \end{aligned}$$

The signal field is (since $(\xi_i^{11})^2 = 1$)

$$\begin{aligned} h_i^s(I^{11}) &= (\xi_i^{11} - m^1 \xi_i^1) \frac{1}{N} \sum_j (\xi_j^{11} - m^1 \xi_j^1) \xi_j^{11} \\ &= \xi_i^{11} (1 - m^1 \xi_i^{11} \xi_i^1) (1 - (m^1)^2), \end{aligned}$$

which yields $h_i^s(I^{11}) \xi_i^{11} \simeq (1 - m^1)(1 - (m^1)^2)$.

The noise field is

$$\begin{aligned} h_i^n(I^{11}) &= \frac{1}{N} \sum_{\mu\nu} (\xi_i^{\mu\nu} - m^1 \xi_i^\mu) \sum_j (\xi_j^{\mu\nu} - m^1 \xi_j^\mu) \xi_j^{11} \\ &= \sum_{\mu\nu} (\xi_i^{\mu\nu} - m^1 \xi_i^\mu) \frac{1}{N} (I^{\mu\nu} \cdot I^{11} - m^1 I^\mu \cdot I^{11}). \end{aligned}$$

It averages out to zero $\bar{h}_i^n = 0$, and may be written as

$$h_i^n(I^{11}) = \frac{1}{N} \sum_j \sum_{\mu\nu} (\xi_i^{\mu\nu} \xi_i^\mu - m^1)(\xi_j^{\mu\nu} \xi_j^\mu - m^1) \xi_i^\mu \xi_j^\mu \xi_j^{11},$$

which is similar to the noise term we saw in the preceding section. Therefore the noise field is given by

$$\overline{(h_i^n)^2} = \frac{P}{N} (1 - (m^1)^2)^2$$

and the memory storage capacity by

$$P_c = N(1 - m^1)^2.$$

This result shows that the memory capacity is fully determined by the overlap m^1 between the patterns of the lowest level of the hierarchical tree. This memory storage capacity however is not that good. There exist other rules which dramatically improve the performance of networks at storing correlated patterns. We examine two such algorithms in the following sections.

7.5.4 An improved learning rule

The existence of inter pattern correlations increases the memory capacity of neural networks and possibly makes it grow beyond any limit as shown by a direct computation of the volume of solutions Γ (see section 7.5.7): therefore there must exist learning rules which present this property of infinite memory storage capacities. Such a model has been devised by Buhmann, Divko and Schulten.

Let the distribution probability of the states of the patterns be

$$P(\xi_i^\mu) = m \delta(1 - \xi_i^\mu) + (1 - m) \delta(\xi_i^\mu), \quad (7.51)$$

which means that $\xi_i^\mu \in \{0, 1\}$: here we work with the 0, 1 representation. The average number of 1's is mN . Patterns are uncorrelated when $m = \frac{1}{2}$. In this model one chooses, naturally enough, the background state as $I^{\text{back}} = \{\xi_i^{\text{back}} = 0\}$: patterns become more and more similar to the background when m shrinks to zero. Moreover, it is necessary to introduce a threshold θ to obtain interesting results. The synaptic efficacies are given by

$$J_{ij} = \frac{1}{N} \sum_{\mu=1}^P (\xi_i^\mu - m)(\xi_j^\mu - m), \quad \text{with } \xi_i^\mu \in \{0, 1\}, \quad (7.52)$$

a formula devised by Buhmann, Divko and Schulten, which looks similar to the one put forward by Amit but which is not because the representations which have been used are different in the two cases. The local

	$\xi_i^\mu \xi_j^\mu \xi_j^1$	$X(C)$	$X^2(C)$	$P(C)$
C	1 1 1	$(1-m)^2$	$(1-m)^4$	m^3
	1 1 0	0	0	$m^2(1-m)$
	1 0 1	$-m(1-m)$	$m^2(1-m)^2$	$m^2(1-m)$
	1 0 0	0	0	$m(1-m)^2$
	0 1 1	$-m(1-m)$	$m^2(1-m)^2$	$m^2(1-m)$
	0 1 0	0	0	$m(1-m)^2$
	0 0 1	m^2	m^4	$m(1-m)^2$
	0 0 0	0	0	$(1-m)^3$

Table 7.2.

field for state I^1 is given by

$$\begin{aligned}
h_i(I^1) &= \sum_j J_{ij} \xi_j^1 - \theta \\
&= \frac{1}{N} \sum_j (\xi_i^1 - m)(\xi_j^1 - m) \xi_j^1 - \theta \\
&\quad + \frac{1}{N} \sum_j \sum_{\mu \neq 1} (\xi_i^\mu - m)(\xi_j^\mu - m) \xi_j^1 \\
&= h_i^s + h_i^n.
\end{aligned}$$

Using the identity $(\xi_j^\mu)^2 = \xi_j^\mu$, one has

$$h_i^s(\xi_i^1) = (\xi_i^1 - m) \frac{1}{N} \sum_j (1-m) \xi_j^1 - \theta = (\xi_i^1 - m)(1-m)m - \theta,$$

whence

$$h_i^s(\xi_i^1 = 1) = (1-m)^2 m - \theta$$

and

$$h_i^s(\xi_i^1 = 0) = -(1-m)m^2 - \theta.$$

We fix the threshold θ right between these two values:

$$(1-m)^2 m - \theta = (1-m)m^2 + \theta.$$

This leads to $\theta = \frac{1}{2}m(1-m)(1-2m)$ and the signal is given by

$$h_i^s(\xi_i^1 = 1) = \frac{1}{2}(1-m)m = -h_i^s(\xi_i^1 = 0).$$

The first and the second moment of the noise are computed from Table 7.2, where $X(C)$ is given by

$$X(C) = (\xi_i^\mu - m)(\xi_j^\mu - m)\xi_j^1.$$

The first moment vanishes,

$$\overline{h_i^n} = \frac{P}{N} \sum_C X(C) P(C) = 0,$$

and the second moment is given by

$$\overline{(h_i^n)^2} = \frac{P}{N} \sum_C X^2(C) P(C) = \frac{P}{N} m^3 (1-m)^2.$$

As usual the memory storage capacity is determined by the condition that the noise amplitude overshoots the signal. This leads to

$$\frac{1}{4}m^2(1-m)^2 = \frac{P_c}{N} m^3(1-m)^2 \quad \text{or} \quad P_c = \frac{N}{4m}. \quad (7.53)$$

This result comes as a surprise since *the memory storage capacity now grows as the patterns become more and more correlated*, that is to say when m descends to zero. In actual fact this result was anticipated some twenty years ago by Willshaw *et al.*, who used a model which we will describe in the next section. As the number of patterns increases when m is low it appears interesting to store patterns comprising small numbers of 1's: this is *sparse coding*. Obviously m cannot dwindle to zero. There must exist an optimal overlap m . This question is also discussed in the next section.

Remarks

a) Similar results are obtained by starting from *an asymmetric connectivity matrix with a presynaptic bias*:

$$J_{ij} = \frac{1}{N} \sum_\mu \xi_i^\mu (\xi_j^\mu - m). \quad (7.54)$$

In this case the noise field averages out to zero

$$\begin{aligned} h_i^n(I^1) &= \frac{1}{N} \sum_\mu \sum_j \xi_i^\mu (\xi_j^\mu - m) \xi_j^1 \\ &= \sum_\mu \xi_i^\mu \frac{1}{N} \sum_j (\xi_j^\mu \xi_j^1 - m \xi_j^1) \\ &= \sum_\mu \xi_i^\mu \left(\frac{1}{N} \sum_j \xi_j^\mu \xi_j^1 - \frac{m}{N} \sum_j \xi_j^1 \right) \\ &= \sum_\mu \xi_i^\mu (m^2 - m^2) = 0, \end{aligned}$$

and the signal-to-noise analysis proceeds as above.

b) One could think that the very same result could be obtained by using *an asymmetric connectivity matrix with a postsynaptic bias*:

$$J_{ij} = \frac{1}{N} \sum_{\mu} (\xi_i^{\mu} - m) \xi_j^{\mu}.$$

This is a pitfall which is revealed by looking at the noise term:

$$\begin{aligned} h_i^n(I^1) &= \frac{1}{N} \sum_{\mu} \sum_j (\xi_i^{\mu} - m) \xi_j^{\mu} \xi_j^1 \\ &= \sum_{\mu} (\xi_i^{\mu} - m) \left(\frac{1}{N} \sum_j \xi_j^{\mu} \xi_j^1 \right) \\ &= \sum_{\mu} (\xi_i^{\mu} - m) m^2 = m^2 \sum_{\mu} (\xi_i^{\mu} - m). \end{aligned}$$

The noise term is a random variable whose distribution does not narrow as the number P patterns increases. The width is of the order of \sqrt{P} and quickly overshoots the signal amplitude.

7.5.5 The Willshaw model

The model of Buhmann *et al.* was inspired by an older model introduced by Willshaw, Buneman and Longuet-Higgins in 1969, in reality well before all models we have seen so far had been devised. The idea is to appeal to the ‘old Hebbian dynamics’ which states, according to Hebb, that a synaptic efficacy J_{ij} is modified when, and only when, neurons i and j are both active. It is assumed moreover that the efficacies are binary variables $J_{ij} \in \{0, 1\}$.

The convenient coding is the 0, 1 representation and the connectivity matrix is defined by

$$J_{ij} = \mathbf{1} \left(\sum_{\mu} \xi_i^{\mu} \xi_j^{\mu} \right), \quad \text{with } \xi_i^{\mu} \in \{0, 1\}, \quad (7.55)$$

and

$$\mathbf{1}(x) = \begin{cases} 1 & \text{if } x > 0, \\ 0 & \text{if } x \leq 0. \end{cases}$$

In order that not all efficacies are saturated to $J_{ij} = 1$ it is necessary that the patterns I^{μ} have a large number of 0’s. In other words m , the proportion of 1’s in the patterns, is assumed to be low (sparse coding). On the other hand, the local fields are positive quantities and a threshold θ , whose value must be carefully determined, is to be introduced.

The local field, when the system is in state I^1 , is given by

$$h_i(I^1) = \sum_j J_{ij} \xi_j^1.$$

The signal term is computed by setting $\mu = 1$ in Eq. (7.55),

$$h_i^s(I^1) = Nm \xi_i^1,$$

since $J_{ij} = 1$ for all j 's such that $\xi_i^1 \xi_j^1 = 1$.

The threshold is set to its maximum value,

$$\theta = Nm.$$

It must be recalled that all quantities entering the definition of the local field are positive and that a spurious neuronal state is a state with $\xi_i^1 = 0$ (which gives a signal $h_i^s = 0$), which is nevertheless forced to $S_i = 1$ by the noise field.

Let $\bar{\omega}$ be the probability of an element J_{ij} being $J_{ij} = 1$. Then the probability of the noise term being $h_i^n = Nm$ is the probability that all the J_{ij} 's with $\xi_j^1 = 1$ are $J_{ij} = 1$. This probability is therefore

$$\bar{\omega}^{Nm}$$

and the number of spurious neuronal states is

$$Nm \bar{\omega}^{Nm}.$$

Let us compute $\bar{\omega}$:

- m^2 is the probability that $\xi_i^\mu \xi_j^\mu = 1$.
- $(1 - m^2)$ is the probability that $\xi_i^\mu \xi_j^\mu = 0$.
- $(1 - m^2)^P$ is the probability that $\sum_\mu \xi_i^\mu \xi_j^\mu = 0$.
- And finally the probability that $J_{ij} = 1$ is given by

$$\bar{\omega} = 1 - (1 - m^2)^P. \quad (7.56)$$

The appearance of a single wrong bit is signed by the equation

$$Nm \bar{\omega}^{Nm} = 1. \quad (7.57)$$

Equations (7.56) and (7.57) give the memory storage capacity of the system. From Eq. (7.56) one obtains

$$P \simeq -\frac{\log(1 - \bar{\omega})}{m^2}, \quad (7.58)$$

and from Eq. (7.57),

$$\bar{\omega} = N^{-1/mN}. \quad (7.59)$$

The memory storage capacity is given by

$$P_c = -\frac{\log(1 - N^{-1/mN})}{m^2}. \quad (7.60)$$

The capacity increases as m becomes smaller and smaller. However, when the number of 1’s dwindle, the information that is contained in a given pattern shrinks. Then, what may be considered as the relevant quantity is the stored information \mathcal{J}^T and this is the quantity one could like to maximize. Optimizing the stored information yields an optimal value for m : the number of patterns one can build with mN bits $\xi_i = 1$ taken out of strings of N bits is $\binom{N}{mN}$ and therefore the information embedded into P such patterns is (see a note on the theory of information at the end of this section)

$$\mathcal{J}^T = P \log \left(\frac{N}{mN} \right) \simeq -PN \left(m \log(m) + (1-m) \log(1-m) \right). \quad (7.61)$$

Using Eqs (7.58) and (7.59) rewritten as

$$\frac{\log(N)}{m} = -N \log(\bar{\omega}),$$

the information \mathcal{J}^T , in the limit of small m ’s, becomes

$$\mathcal{J}^T = \frac{N^2}{\log(N)} \log(\bar{\omega}) \log(1-\bar{\omega}) (1-\log(m)). \quad (7.62)$$

$\log(m)$ is of the order of $-\log(N)$ and \mathcal{J}^T is maximal for $\bar{\omega} = \frac{1}{2}$ (one synaptic efficacy J_{ij} in two is 1, the other is 0). Then

$$m = -\frac{\log(N)}{N \log(\bar{\omega})} = \frac{\log(N)}{N \log 2},$$

which means that each pattern comprises a number K of bits with K given by

$$K = Nm = \frac{\log(N)}{\log 2}.$$

Finally, the memory capacity P_c is

$$P_c = \frac{\log 2}{m^2} = \left(\frac{N}{\log(N)} \right)^2 (\log 2)^3,$$

which could be written as

$$P_c = \alpha_c N, \quad \text{with} \quad \alpha_c = \frac{N}{(\log(N))^2} (\log 2)^3.$$

7.5.6 Entropy and information

The *entropy* S^T of a system obeying a set of constraints \mathcal{C} is the logarithm of the number $\Gamma(\mathcal{C})$ of states I which are compatible with \mathcal{C} .

$$S^T(\mathcal{C}) = \log(\Gamma(\mathcal{C})).$$

For example the system is made of N units and every unit has q internal states. A configuration \mathcal{C} is any state I such that N_1 units are in state $q = 1$, N_2 in state $q = 2$, and so on. The number of configurations \mathcal{C} is given by

$$\Gamma(\mathcal{C}) = \frac{N!}{\prod_q N_q!},$$

since there are $N!$ ways of choosing N objects and since a configuration obeying \mathcal{C} is not modified by any permutation of objects with the same label q . Using the Stirling formula, $\log(N!) \simeq N(\log(N) - 1)$, one finds

$$\log(\Gamma(\mathcal{C})) \simeq N \log(N) - \sum_q N_q \log(N_q) = - \sum_q N_q \log\left(\frac{N_q}{N}\right),$$

where the equality $N = \sum_q N_q$ has been used. The entropy per unit S is

$$S = \frac{S^T}{N} = - \sum_q p_q \log(p_q), \quad \text{with} \quad p_q = \frac{N_q}{N}.$$

When the number of internal states is reduced to two states, namely $S_i = 0$ and $S_i = 1$, the entropy per unit is

$$S(m) = -(m \log(m) + (1-m) \log(1-m)),$$

where mN is the numbers of 1's in a pattern.

The definition of *information* J^T is related to the number of bits which are necessary to fully describe a configuration \mathcal{C} . Let us assume that we have a set of states $I = \{S_1, S_2, S_3\}$ that one wants to transmit through a line. If the states I are made of seemingly uncorrelated unit states S_i such as $\{0, 0, 1\}$, $\{1, 0, 1\}$, $\{1, 1, 0\}, \dots$, the number of relevant bits to be transmitted at every message is three. However, if the strings are always either $\{0, 0, 0\}$ or $\{1, 1, 1\}$ it is enough to send one-bit messages. as the other bits can be inferred from the transmitted element of information only. The number of bits that are necessary is therefore given by

$$K = \frac{\log(\Gamma(\mathcal{C}))}{\log 2},$$

where $\Gamma(\mathcal{C})$ is the number of messages (of states) belonging to the type \mathcal{C} of messages which are currently transmitted: $8 = 2^3$ states in the first instance and $2 = 2^1$ in the second. Information is defined by

$$\mathcal{J}^T = K \log 2 = \log(\Gamma(\mathcal{C})).$$

Once it is realized that the meanings of $\Gamma(\mathcal{C})$ are strictly the same from the entropy point of view as they are from the information point of view, one concludes that entropy and information are two facets of a unique concept. One has

$$\mathcal{J}^T = S^T \quad \text{or} \quad \mathcal{J} = \frac{\mathcal{J}^T}{N} = S,$$

where \mathcal{J} is the information per unit. Sometimes information is defined as the opposite of the entropy. The minus sign comes from a change of perspective between the emitter and the receiver of a message. The receiver observes a series of unit states transmitted through the line. Let us assume that K is small. Then it is easy for the receiver to infer the states which are likely to be uttered in forthcoming messages. Information brought about by the observation of highly structured messages is therefore low. The point of view of the emitter is different: for him the more ordered the message, the greater the amount of information he has to pour in it. Small K 's correspond to embed much information in the messages to be sent, large K 's mean that the message source generates the unit states at random. This is the emitter point of view which is often adopted in the theory of information, whence the minus sign.

7.5.7 Computing the volume of solutions for correlated patterns

The relative volume of the space of interactions which stabilize correlated patterns has been calculated by E. Gardner. The computational technique is similar to the one displayed in detail in section 6.3 except that the distribution

$$P(\xi_i^\mu) = \frac{1}{2} \delta(\xi_i^\mu - 1) + \frac{1}{2} \delta(\xi_i^\mu + 1)$$

is replaced by the distribution

$$P(\xi_i^\mu) = \frac{1}{2} (1 + m) \delta(\xi_i^\mu - 1) + \frac{1}{2} (1 - m) \delta(\xi_i^\mu + 1).$$

Here $m = 0$ corresponds to uncorrelated patterns and $m \rightarrow 1$ corresponds to states highly correlated with a background state

$$I^{\text{back}} = \{\xi_i^\mu = +1\}.$$

The calculations follow the main lines which have been detailed in section 6.3. The existence of correlations introduces a new order parameter M , defined by $M = N^{-1/2} \sum_j J_j$, which makes the derivations somewhat more involved. The final equation (6.21),

$$1 = \alpha_c(\kappa) \int_{-\kappa}^{+\infty} Dy (y + \kappa)^2, \quad \text{with} \quad Dy = dy \exp(-\frac{1}{2} y^2),$$

is to be replaced by

$$1 = \alpha_c(m, \kappa) \sum_{\sigma \in \{+1, -1\}} \frac{1}{2} (1 + \sigma m) \int_{(\sigma v m - \kappa)/\sqrt{1-m^2}}^{+\infty} Dy \left[y + \frac{\kappa - \sigma v m}{\sqrt{1-m^2}} \right]^2,$$

where v is the solution of

$$\sum_{\sigma \in \{+1, -1\}} \frac{1}{2} (1 + \sigma m) \int_{(\sigma v m - \kappa)/\sqrt{1-m^2}}^{+\infty} Dy \left[y + \frac{\kappa - \sigma v m}{\sqrt{1-m^2}} \right] = 0. \quad (7.63)$$

The memory storage capacity increases when the correlation m between the patterns to be memorized increases in accordance with the results of previous sections. For unbiased stabilization criterions $\kappa = 0$ one has

$$\frac{P_c}{N} = \alpha_c = 2 \left(1 + \frac{2}{\pi^2} m^2 \right)$$

for small values of m and

$$\alpha_c = \frac{-1}{(1-m) \log(1-m)}$$

for m close to $m = 1$ (see Fig. 7.13).

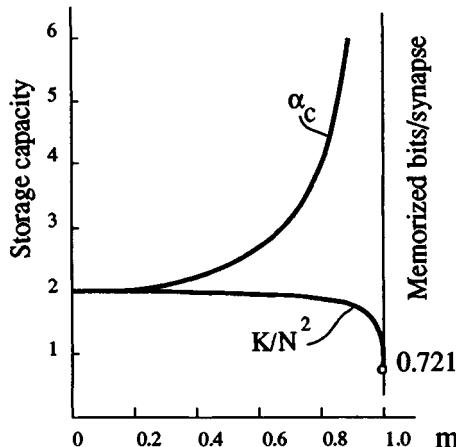


Figure 7.13. Maximum memory storage capacity of neural networks with correlated patterns whose overlap with the background is m (After Gardner). K^2/N is the number of stored bits per existing synaptic link.

The amount of stored information is

$$\mathcal{J}^T = P_c(\text{patterns}) \times N(\text{bits per pattern}) \times \mathcal{J}(\text{information per bit}),$$

and therefore the number of effective bits is (see the note on information in the preceding section)

$$K = \frac{\mathcal{J}^T}{\log 2} = -\frac{N^2}{\log 2} \alpha_c(m) \left[\frac{1}{2}(1-m) \log \frac{1}{2}(1-m) + \frac{1}{2}(1+m) \log \frac{1}{2}(1+m) \right].$$

In the limit $m = 0$, that is to say for uncorrelated patterns one finds

$$K = 2N^2,$$

which means that the system stores 2 bits per synaptic link. It is interesting to note that this result does not depend on the range of synaptic weights. Storing uncorrelated patterns on 32 bits synaptic efficacies instead of 16 bits ones for example does not increase much the amount of information that the network is able to memorize. Conversely, one can imagine that the capacity is not deteriorated much if one restricts the range of efficacies. In the extreme case of binary synapses ($J_{ij} \in \{+1, -1\}$) Mézard has shown that $\alpha_c = 0.83$.

In the limit $m \rightarrow 1$, one finds

$$K = \frac{N^2}{2 \log 2} = 0.721 N^2,$$

which shows that the existence of correlations slightly lowers the maximum amount of stored information and that the effect is not catastrophic.

M. Virasoro goes a step further by applying Gardner’s calculations to *hierarchically organized patterns*, more specially to patterns $I^{\mu\nu}$ obeying the distribution given by Eq. (7.50) with $m^0 = 0$. μ labels the (orthogonal) classes and ν labels the individuals of a given class. For a given class one distinguishes the (+) sites where $\xi_i^\mu = +\xi_i^{\mu\nu}$ from the (–) sites where $\xi_i^\mu = -\xi_i^{\mu\nu}$. One can say that the first type of site confirms the class μ and that the second characterizes the individuals $\mu\nu$. The calculation yields the average depths of basins of attractions measured on both types of sites, that is the distributions of quantities,

$$x_i^{\mu\nu} = \xi_i^{\mu\nu} \sum_j J_{ij} \xi_j^{\mu\nu},$$

measured for each pattern $I^{\mu\nu}$ on sites with $\xi_i^\mu = +\xi_i^{\mu\nu}$ on the one hand and on sites with $\xi_i^\mu = -\xi_i^{\mu\nu}$ on the other and averaged over all possible patterns $I^{\mu\nu}$ obeying the distribution (7.50).

Virasoro finds that the distribution $F^-(x)$ of depths on ‘individual’ sites is much more concentrated at small depths than the distribution $F^+(x)$ of depths on ‘class’ sites. For example, he finds

$$\overline{F^+} \simeq 4 \times \overline{F^-}$$

for $m^1 = 0.6$ close to the maximum storage capacity. This means that moderate levels of noise or attritions of some synapses destabilize those neurons which characterize the individuals, whereas the others remain stabilized: the system is still able to retrieve the classes μ but it is unable to retrieve the individuals $\mu\nu$. This syndrome is characteristic of a perception disorder called prosopagnosia. Prosopagnosia is an impairment in face recognition of individuals while faces are still recognized as faces. Actually, it has been observed that the syndrome is accompanied by a more general difficulty in distinguishing the individuals of a class whatever the class, be it the class of faces or that of cars, whereas the classes themselves are well categorized. This observation gives a clue regarding the way hierarchically correlated patterns are stored. If they are stored according to the procedure P1 (see section 7.5.3) there exist neurons which code for a specific class and others which code for individuals of that class. If such a case arises, prosopagnosia can be brought about only by the destruction of the latter population of neurons, i.e. by localized lesions. If the patterns are stored according to procedure P2, prosopagnosia can arise by diffuse lesions. This seems to be the case.

SOLVING THE PROBLEM OF CREDIT ASSIGNMENT

The architectures of the neural networks we considered in Chapter 7 are made exclusively of visible units. During the learning stage, the states of *all neurons* are entirely determined by the set of patterns to be memorized. They are so to speak pinned and the relaxation dynamics plays no role in the evolution of synaptic efficacies. How to deal with more general systems is not a simple problem. Endowing a neural network with hidden units amounts to adding many degrees of freedom to the system, which leaves room for ‘*internal representations*’ of the outside world. The building of learning algorithms that make general neural networks able to set up efficient internal representations is a challenge which has not yet been fully satisfactorily taken up. Pragmatic approaches have been made, however, mainly using the so-called back-propagation algorithm. We owe the current excitement about neural networks to the surprising successes that have been obtained so far by calling upon that technique: in some cases the neural networks seem to extract the unexpressed rules that are hidden in sets of raw data. But for the moment we really understand neither the reasons for this success nor those for the (generally unpublished) failures.

8.1 The back-propagation algorithm

8.1.1 A direct derivation

To solve the credit assignment problem is to devise means of building relevant internal representations; that is to say, to decide which state $I^{\mu, \text{hid}}$ of hidden units is to be associated with a given pattern $I^{\mu, \text{vis}}$ of visible units. Several schemes have been proposed so far. The most natural approach is to assume that the hidden states are unambiguously determined by the input states and the output states by the hidden states through the usual dynamics of neuronal states. The architecture is then necessarily feedforward. The synaptic connections are modified so as to make the observed outputs closer and closer to the wanted outputs: learning is therefore an error-correction process. Feedforward

architectures and error correction processes make the essence of the *back-propagation algorithm*. The internal representations that the algorithm builds are states of hidden units $I^{\mu,\text{hid}}$, which are triggered by the input states $I^{\mu,\text{in}}$ once the learning process arrives at a fixed point. This approach, contrary to that put at work by Fukushima in his Neo-Cognitron for example, does not appeal to any preknowledge of the form that internal representations should take.

Feedforward networks are characterized by their lack of oriented loops of interactions. The units of a feedforward network can be grouped into classes or *layers* ℓ . A neuron belongs to layer $\ell + 1$ if the length of the *longest* oriented path that links one of the input neurons to that neuron is ℓ . All input neurons belong to layer $\ell = 1$. The state of a unit of layer ℓ is determined by those of units belonging to layers of lower ranks up to input layer and therefore information necessarily flows from the top down through all the network, so avoiding the system having to learn about itself (Fig. 8.1).

The average activity σ_i of neuron i is given by

$$\sigma_{i \in \ell} = \mathcal{S}\left(\beta \sum_{j \in \{\ell\}} J_{ij} \sigma_j\right), \quad (8.1)$$

where $\{\ell\}$ is the set of layers m such as $m < \ell$ and \mathcal{S} the response function of neurons. For the sake of simplicity, σ_i stands here for $\langle \sigma_i \rangle$, the usual notation for the average activity of neuron i . Let us first consider strictly feedforward layered networks wherein neurons of layers $(\ell - 1)$ connect only to neurons of layer ℓ .

Let $I^{\mu,\text{in}} = \{\xi_{m \in \ell=1}^\mu\}$ and $I^{\mu,\text{out}} = \{\xi_{i \in \ell=L}^\mu\}$, with $\mu = 1, 2, \dots, P$, be the input patterns and the output patterns that the system has to associate. The states ξ^μ , be they input or output states, may be *continuous variables* $\xi^\mu \in [-1, +1]$.

When the network experiences the input pattern $I^{\mu,\text{in}}$ the activities of neurons belonging to layer $(L - 1)$ are given by the following formula:

$$\sigma_{j \in L-1}(I^\mu) = \mathcal{S}\left[\beta \sum_{k \in L-2} J_{jk} \mathcal{S}\left[\beta \sum_{h \in L-3} J_{kh} \mathcal{S}\left[\beta \cdots \left[\sum_{m \in \ell=1} J_{hm} \xi_m^\mu\right]\right]\right]\right].$$

Following the development of section 7.4, the better is the response of the system to the input pattern $I^{\mu,\text{in}}$, the lower is the cost function H . According to Eq. (7.28) the cost function is given by

$$H = \sum_{i \in L} \sum_{\mu}^P g(x_i^\mu),$$

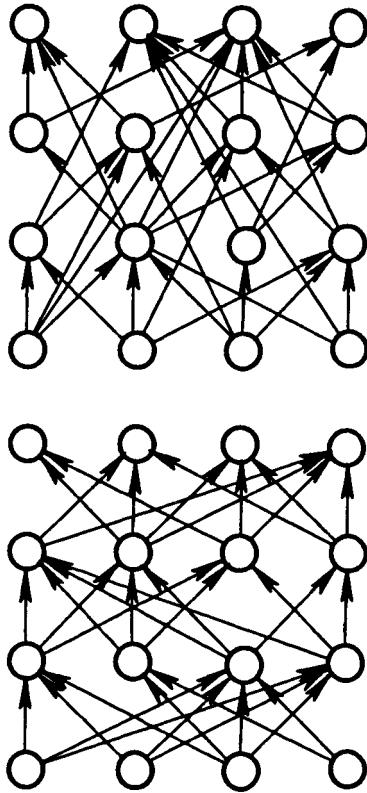


Figure 8.1. Feedforward and strictly feedforward layered networks.

where the stabilization parameters x_i^μ are

$$x_i^\mu = \xi_i^{\mu, \text{out}} h_i^\mu, \quad i \in L, \quad \text{and} \quad h_{i \in L}^\mu = \sum_{j \in L-1} J_{ij} \sigma_j(I^\mu).$$

The learning rule is determined by the gradient algorithm Eq. (7.22),

$$\frac{dJ_{ij}}{dt} = -\varepsilon \frac{\partial H}{\partial J_{ij}}, \quad \varepsilon > 0,$$

which makes sure that the cost is a non-increasing function of time. The cost is explicitly given by

$$H = \sum_{\mu}^P \sum_{i \in L} g \left[\xi_i^{\mu, \text{out}} \sum_{j \in L-1} J_{ij} \sigma_j(I^\mu) \right]$$

and $\Delta J_{i \in L, j \in L-1} = -\varepsilon \sum_{\mu=1}^P [f(\xi_i^{\mu, \text{out}} h_i^\mu) \xi_i^{\mu, \text{out}}] \sigma_j(I^\mu)$,

with $f(x) = \frac{dg(x)}{dx}$.

Let us now compute the variations of interactions connecting layer $(L-2)$ to layer $(L-1)$. One considers the full formula for the cost function,

$$H = \sum_{\mu}^P \sum_{i \in L} g \left[\xi_i^{\mu, \text{out}} \sum_{j \in L-1} J_{ij} S \left(\beta \sum_{k \in L-2} J_{jk} \sigma_k(I^\mu) \right) \right],$$

whence

$$\begin{aligned} \Delta J_{j \in L-1, k \in L-2} &= -\varepsilon \sum_{\mu}^P \sum_{i \in L} [f(\xi_i^{\mu, \text{out}} h_i^\mu) \xi_i^{\mu, \text{out}}] J_{ij} \beta \sigma_k(I^\mu) T(\beta h_j^\mu), \end{aligned}$$

with $T(x) = \frac{dS(x)}{dx}$.

One introduces new notations,

$$Y_{i \in L}(I^\mu) = f(\xi_i^{\mu, \text{out}} h_i^\mu) \xi_i^{\mu, \text{out}},$$

$$\begin{aligned} Y_{j \in L-1}(I^\mu) &= \beta T(\beta h_j^\mu) \sum_{i \in L} J_{ij} [f(\xi_i^{\mu, \text{out}} h_i^\mu) \xi_i^{\mu, \text{out}}] \\ &= \beta T(\beta h_j^\mu) \sum_{i \in L} J_{ij} Y_{i \in L}(I^\mu), \end{aligned}$$

so as to rewrite the variations of interactions as

$$\Delta J_{i \in L, j \in L-1} = -\varepsilon \sum_{\mu}^P Y_i(I^\mu) \sigma_j(I^\mu),$$

$$\Delta J_{i \in L-1, j \in L-2} = -\varepsilon \sum_{\mu}^P Y_i(I^\mu) \sigma_j(I^\mu).$$

The computation is extended to all layers:

$$\Delta J_{i \in \ell, j \in \ell-1} = -\varepsilon \sum_{\mu}^P Y_i(I^\mu) \sigma_j(I^\mu), \quad (8.2)$$

$$Y_{i \in \ell}(I^\mu) = \beta T(\beta h_i^\mu) \sum_{k \in \ell+1} J_{ki} Y_k(I^\mu). \quad (8.3)$$

Taking for f the form used in the perceptron algorithm, that is to say

$$f(\xi_i^{\mu,\text{out}} h_i^\mu) = -\frac{1}{2}(1 - \text{sign}(\xi_i^{\mu,\text{out}} h_i^\mu)),$$

one finds that

$$\begin{aligned} Y_{i \in L} &= -\frac{1}{2}(1 - \text{sign}(\xi_i^{\mu,\text{out}} h_i^\mu)) \xi_i^{\mu,\text{out}} \\ &= -\frac{1}{2}(\xi_i^{\mu,\text{out}} - \text{sign}(h_i^\mu)) = \frac{1}{2}(\xi_i^{\mu,\text{out}} - \sigma_i(I^\mu)), \end{aligned} \quad (8.4)$$

assuming that β^{-1} tends to zero at least for the last layer.

Then $Y_{i \in L}$ is none other than the difference between the wanted output $\xi_i^{\mu,\text{out}}$ and the observed one $\sigma_i(I^\mu)$. The cost function is the Hamming distance between the desired output state $I^{\mu,\text{out}}$ and the observed output state $\{\sigma_i(I^\mu)\}$. Equations (8.1) to (8.4) define the *back-propagation algorithm* (see Fig. 8.2). It is interesting to note that the formulae remain valid for general feedforward networks. The back-propagation algorithm is displayed next page.

The back-propagation algorithm has been proposed independently by a number of authors, Le Cun and Rumelhart in particular. It is probably the most popular learning rule to date. It is able to find sets of connections and thresholds which implement the XOR function even in the minimal strictly feedforward network, i.e. that which comprises two neurons in a unique hidden layer. Therefore it overcomes the difficulty emphasized by Minsky and Papert, of simple neural networks not being able to implement non-linearly separable functions (this property is shared by other algorithms which are presented in the next sections). The most serious drawback is that no convergence theorem is known for the rule, so that success or failure, when applied to a specific application, seems to be mainly a matter of chance.

What the learning rule actually does is well illustrated by an example which has been studied by Lang and Witbrock. This is the learning of two intermingled spirals of points. The network is comprised of two continuous input units, so that the input space is the $[-1, +1]^2$ square, of two hidden layers of five units each, and of a single binary output unit which must be $\sigma^{\text{out}} = +1$ if the activity (ξ_1, ξ_2) of the input units represents a point of the first spiral and $\sigma^{\text{out}} = -1$ if it represents a point of the other spiral (see Fig. 8.3). One sees, in the diagrams which display the activities of the various hidden units, how the simple linear discrimination carried out by the first layer becomes more involved in the second layer so as to produce the right regions of discrimination in the output unit.

Let $\{i^-\}$ be the fan-in set of units, including the threshold neuron, which directly feeds the neuron i and $\{i^+\}$ the fan-out set of neurons which are directly fed by neuron i (see Fig. 8.2).

- 1) For all patterns I^μ , compute the fields

$$h_i^\mu = \sum_{j \in \{i^-\}} J_{ij} \sigma_j(I^\mu)$$

and all activities of the network

$$\sigma_i(I^\mu) = \mathcal{S}(\beta h_i^\mu)$$

This computation must be carried out one layer after the other starting with $\sigma_{j \in \ell=1}(I^\mu) = \xi_j^{\mu, \text{in}}$ down to the last layer $\ell = L$.

- 2) Compute the learning parameters $Y_i(I^\mu)$ according to

$$Y_i(I^\mu) = \beta T\left(\beta h_i^\mu\right) \sum_{j \in \{i^+\}} J_{ji} Y_j(I^\mu).$$

Note the reversal of the indices of the \tilde{J} matrix. The computation is carried out one layer after the other, starting from the last layer ($\ell = L$), where

$$Y_{i \in L}(I^\mu) = f(\xi_i^{\mu, \text{out}} h_i^\mu) \xi_i^{\mu, \text{out}}$$

up to the second layer ($\ell = 2$).

- 3) Modify the synaptic efficacies according to a Hebbian-type rule:

$$\Delta J_{ij} = -\varepsilon \sum_\mu^P Y_i(I^\mu) \sigma_j(I^\mu).$$

- 4) Iterate in 1).

The back-propagation algorithm.

8.1.2 Back-propagation as an optimization problem (Le Cun)

Let us start with the error function

$$H(I^\mu) = \frac{1}{4} \sum_{i \in L} (\sigma_i(I^\mu) - \xi_i^{\mu, \text{out}})^2$$

as the cost function. The cost function has to be minimized with respect to the synaptic efficacies J_{ij} under the constraint that the rules of the neuronal dynamics must be obeyed:

$$\varphi_{i \in \ell}(I^\mu) = \sigma_i(I^\mu) - \mathcal{S}(\beta h_i^\mu) = 0.$$

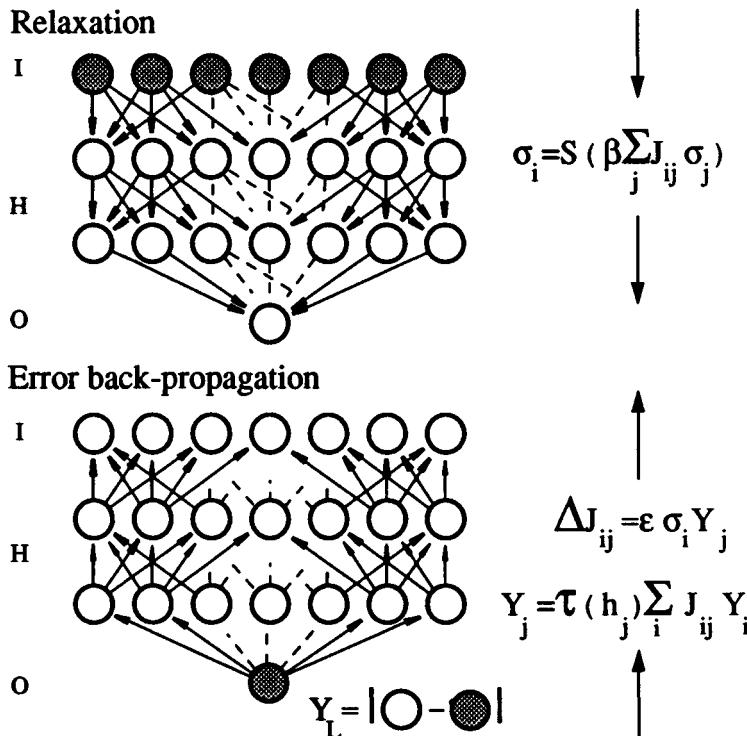


Figure 8.2. Forward propagation of signals and backward propagation of errors in the back-propagation algorithm.

One therefore introduces Lagrange multipliers z_i^μ , and the cost function which is to be minimized becomes

$$\mathcal{H}(I^\mu) = H(I^\mu) + \sum_i z_i^\mu \varphi_i(I^\mu).$$

This is carried out by using a gradient dynamics:

$$\Delta J_{ij}(I^\mu) = -\epsilon \frac{\partial \mathcal{H}(I^\mu)}{\partial J_{ij}}, \quad \epsilon > 0,$$

with $\frac{\partial \mathcal{H}(I^\mu)}{\partial \sigma_i} = 0$ and $\frac{\partial \mathcal{H}(I^\mu)}{\partial z_i^\mu} = 0$. The last equations give

$$\sigma_i(I^\mu) = S(\beta h_i^\mu),$$

which simply depicts the dynamics for the neuronal states. With the second condition one obtains

$$z_{i \in L}^\mu = \frac{1}{2} (\xi_i^{\mu, \text{out}} - \sigma_i(I^\mu)), \quad \text{or} \quad z_{i \in \ell}^\mu = \beta \sum_{j \in \ell+1} z_j^\mu J_{ji} T(\beta h_j^\mu), \quad \text{if } \ell \neq L,$$

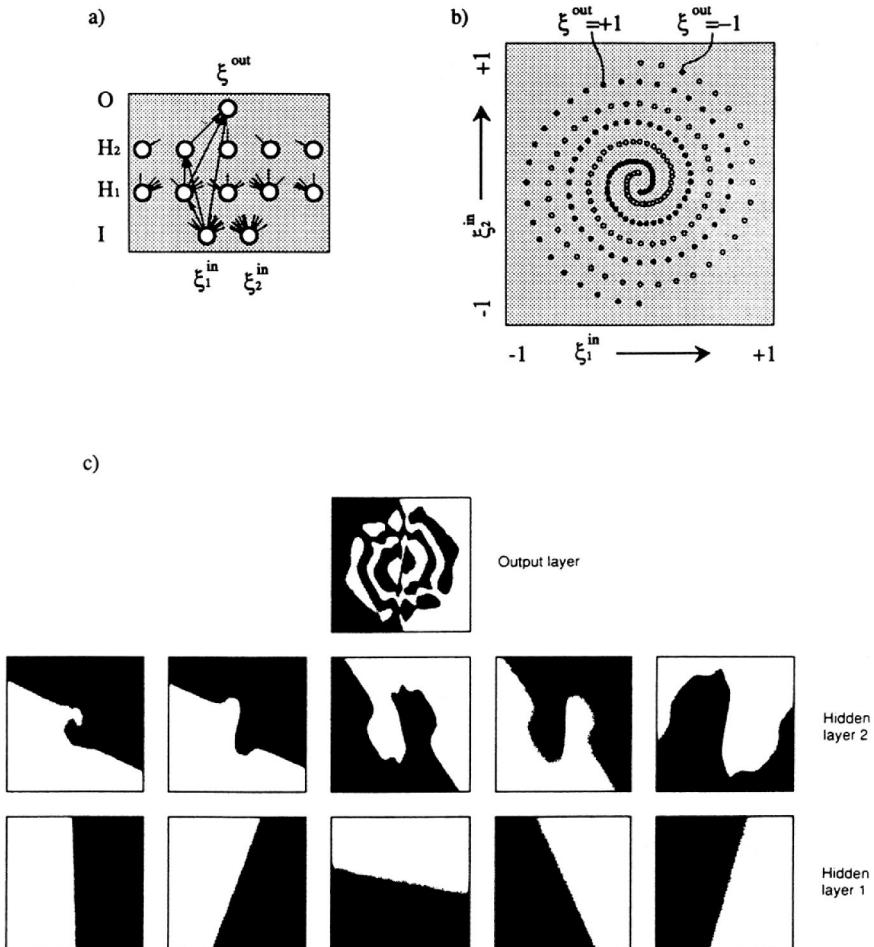


Figure 8.3. An example of back-propagation training.

a) The network. b) The training set. c) The neuronal activities as the inputs range over the unit square (After Lang and Witbrock).

and finally the first derivative gives

$$\Delta J_{i \in \ell, j \in \ell-1} = -\varepsilon \frac{\partial \mathcal{H}}{\partial J_{ij}} = \varepsilon \beta z_i^\mu \sigma_j T(\beta h_i^\mu).$$

With the definitions

$$Y_i(I^\mu) = \begin{cases} -\beta z_i^\mu T(\beta h_i^\mu), & \text{if } i \in \ell \notin L, \\ -\frac{1}{2} (\xi_i^{\mu, \text{out}} - \sigma_i(I^\mu)) & \text{if } i \in L, \end{cases}$$

the back-propagation algorithm is recovered.

8.1.3 Improvements of gradient algorithms

The gradient algorithms suffer from various defects.

- a) One is the *overshooting effect*. If the parameter ε is too small the motion of \tilde{J} will be on the point of ceasing before the cost function reaches its minimum value. If it is too large, \tilde{J} trespasses the target, then goes back and begins to oscillate.

There exist two remedies (at least) for this defect.

The first consists in *modifying the parameter ε* according to the following modulated gradient algorithm.

Let ν be the ν th learning step. Then

$$J_{ij}(\nu + 1) = J_{ij}(\nu) - \varepsilon(\nu + 1) \frac{\partial H}{\partial J_{ij}},$$

where the parameter $\varepsilon(\nu + 1)$ is given by

$$\varepsilon(\nu + 1) = \begin{cases} (1 + \alpha)\varepsilon(\nu) & \text{if } H(\nu + 1) < H(\nu), \\ (1 - \alpha)\varepsilon(\nu) & \text{if } H(\nu + 1) > H(\nu), \end{cases}$$

with ε and $\alpha > 0$.

The modulated gradient algorithm.

The other algorithm is called *the momentum gradient algorithm*. It consists of adding a friction term to the dynamics of interactions \tilde{J} :

$$J_{ij}(\nu + 1) = (1 - \varepsilon) J_{ij}(\nu) - \varepsilon \frac{\partial H}{\partial J_{ij}} \quad \text{with } 0 < \varepsilon < 1.$$

- b) *The gully effect.* This appears when the cost landscape is very asymmetrical in the vicinity of the optimal solution. The above algorithms allow the bottom of the gully to be quickly reached but the motion is still very slow along the gully. It is possible to tackle this problem by appealing to *the projection gradient algorithm* (see next page).

The trajectory of \tilde{J} is made of pieces of orthogonal segments and it seems well suited therefore to the gully problem. For convex problems the projection algorithm is extremely efficient.

- c) Unfortunately, the cost function is not convex in general. The landscape is made of a number of valleys. It is not certain that the present vector \tilde{J} lies in the right valley and there is no more indication as regards the direction the vector has to be moved to improve the cost function.

1) For all i, j compute $\frac{\partial H}{\partial J_{ij}}$.

2) Find $\varepsilon(\nu + 1)$ such that

$$H \left[\left(J_{ij}(\nu) - \varepsilon(\nu + 1) \frac{\partial H}{\partial J_{ij}} \right) \right]$$

is minimum.

3) Then

$$J_{ij}(\nu + 1) = J_{ij}(\nu) - \varepsilon(\nu + 1) \frac{\partial H}{\partial J_{ij}}$$

for all synapses.

4) Iterate in 1).

The projection gradient algorithm .

One way out is to use the thermal annealing procedure; that is, to move \tilde{J} at random and to keep the change with a probability which depends on the variation of the cost function brought about by the change. This is a very slow procedure since the number of dynamical variables, that is the number of interactions, increases as the square of the size of the network.

Another algorithm consists in sampling the space of interactions by using a number of starting states and by appealing to one of the above gradient algorithms. The corresponding metastable vectors \tilde{J}^* are collected and the one with the minimal cost function is kept.

8.2 Handling internal representations

8.2.1 Learning by choice of internal representations (CHIR)

In Chapter 7 we introduced an efficient learning dynamics for systems not comprising hidden units, namely the perceptron algorithm. Obviously it is tempting to apply the perceptron recipe to layered systems. But for the perceptron rule to be applicable it is necessary to know the states of *all neurons* for the set of training patterns, in particular the states of hidden units. In other words it is necessary to know the internal representation of the network.

Using the back-propagation algorithm automatically sets up an internal representation. We have seen in section 8.1.1 how useful the examination of the internal representations of such trained networks is in gaining insights into what learning really achieves. However, it is not

certain that the representations that are obtained that way can be learned by the perceptron algorithm, all the more so since the perceptron algorithm must find a solution both between the input layer and the hidden layer and between the hidden layer and the output layer. One idea is to handle the internal representation in order to find a set of patterns $I^{\mu, \text{hid}}$ which fulfil these requirements.

Following this line of thought, Grossmann, Meir and Domany consider the states $\xi_j^{\mu, \text{hid}}$, $j = 1, \dots, N_H$, $\mu = 1, \dots, P$ of the internal representation as a set of dynamical variables that they are free to change so as to make them more suited to the perceptron algorithm. The problem is to associate through a three-layer, feedforward network (one hidden layer), a set of P input patterns,

$$I^{\mu, \text{in}} = \{\xi_k^{\mu, \text{in}}\}, \quad k = 1, 2, \dots, N_I, \quad \mu = 1, 2, \dots, P,$$

with a set of P output patterns,

$$I^{\mu, \text{out}} = \{\xi_i^{\mu, \text{out}}\}, \quad i = 1, 2, \dots, N_O.$$

The usual equations of the neuronal dynamics give a first internal representation which is modified 'by hand' if it is not satisfactory. The algorithm of learning by choice of internal representations (CHIR) which they put forward, is displayed next page (see also Fig. 8.4).

The algorithm may be modified so as to apply to networks made of any number of hidden layers. It proves to be both faster and more efficient than the back-propagation algorithm in all problems for which both learning rules have been checked. The comparison carried out by Domany *et al.* focuses on categorization problems which require networks with unique output unit $N_O = 1$. The first problem to be studied is the n -contiguity problem, which involves deciding whether the input patterns $I^{\mu, \text{in}}$ have more than n contiguous blocks of +1s. If such is the case then $\sigma^{\mu, \text{out}} = +1$, otherwise $\sigma^{\mu, \text{out}} = -1$. The second is the symmetry problem, in which the state of the output unit $I^{\mu, \text{in}}$ must be $\sigma^{\mu, \text{out}} = +1$ if the input pattern is symmetrical around its center and $\sigma^{\mu, \text{out}} = -1$ otherwise. The last problem is the parity problem, in which one requires that $\sigma^{\mu, \text{out}} = +1$ if the number of +1s in the input pattern $I^{\mu, \text{in}}$ is even and $\sigma^{\mu, \text{out}} = -1$ if this number is odd. In all these problems convergence has been achieved using the CHIR algorithm, which has not been the case with the back-propagation algorithm. Moreover, the time necessary for the CHIR algorithm to converge is much shorter.

8.2.2 The tiling algorithm

The learning algorithms which have been described up to now strive to modify the parameters of a *given* network to make it respond as expected.

Start from a given set of synaptic efficacies $J_{j \in 2, k \in 1}$ (including the thresholds $J_{j,0}$) from layer 1 to layer 2 and $J_{i \in 3, j \in 2}$ (including the thresholds $J_{k,0}$) from layer 2 to layer 3.

- 1) Build the $P \times N_H$ table of internal representation whose elements $\xi_j^{\mu, \text{hid}}$ are given by

$$\xi_j^{\mu, \text{hid}} \equiv \sigma_{j \in 2}(I^\mu) = \text{sign}\left(\sum_{k \in 1}^{N_I} J_{jk} \xi_k^{\mu, \text{in}}\right).$$

- 2) Modify the efficacies $J_{i \in 3, j \in 2}$ of interactions between layer 2 and layer 3 according to the perceptron algorithm. If this stage is successful, that is if the output patterns are associated with the internal representation without any error, the algorithm stops. Otherwise the algorithm is interrupted after a given number of learning steps. At this point one obtains a network whose performance may be characterized by the number of errors it produces:

$$H = \sum_{\mu}^P H(I^\mu) = \sum_{\mu}^P \sum_{i \in 3}^{N_O} (\sigma_i(I^\mu) - \xi_i^{\mu, \text{out}})^2,$$

with $\sigma_{i \in 3}(I^\mu) = \text{sign}\left(\sum_{j \in 2}^{N_H} J_{ij} \xi_j^{\mu, \text{hid}}\right).$

- 3) If $H \neq 0$ modify the table of internal representation while keeping the interactions $J_{i \in 3, j \in 2}$ until the number of errors H possibly vanishes. A strategy for the modification is to look at the table row by row and to change the states in one row (corresponding to a given state $I^{\mu, \text{hid}}$) by starting from states which yield the largest decrease of the error function H . This is carried out for all rows of the table until no further improvement of the cost functions is observed.
- 4) Use the perceptron rule to the weights $J_{j \in 2, k \in 1}$ of synapses connecting the layer 1 to layer 2. If this learning stage is successful, the algorithm stops, otherwise it is interrupted after a given number of steps and,
- 5) Iterate in 1): The efficacies $J_{j \in 2, k \in 1}$ found in step 4) determine a new table of internal representation and the process is iterated until convergence is achieved.

The CHIR algorithm.

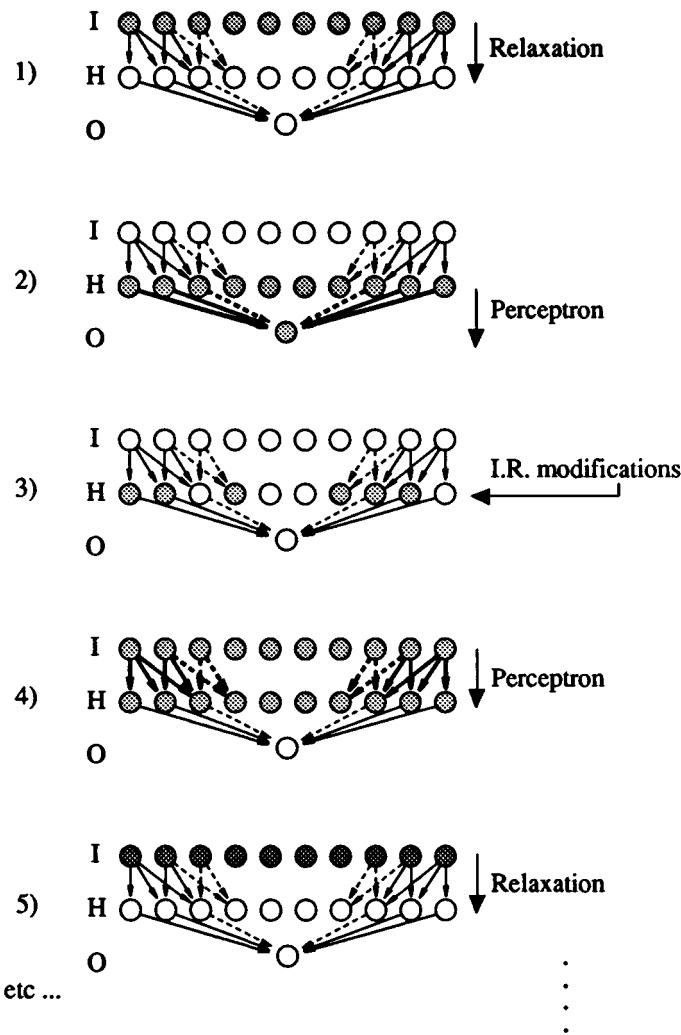


Figure 8.4. The various steps of the CHIR algorithm.

The tiling algorithm (see next page), devised by Mézard and Nadal, however, is of a *constructive nature*. The architecture of the network is a feedforward architecture but neither the number of layers nor the number of units in a given layer are prescribed in advance. Units and layers of the network are added till the system implements the wanted Boolean mapping (which may be a Boolean function as well). Moreover,

and unlike the back-propagation or the CHIR algorithms, convergence is guaranteed.

- The construction proceeds layer after layer. Layer $\ell = 1$ is made of the N_1 input units.
- 1) Use the perceptron algorithm applied to the whole training set, to build the connections between the neurons of layer ℓ and a neuron, the first of layer $\ell + 1$, which tentatively is the output of the network. If this learning stage is successful, that is to say if the number of errors $e_\ell = 0$, construction is achieved. If the perceptron algorithm does not succeed, if $e_\ell \neq 0$, it is stopped when e_ℓ is stable. Then determine the two classes C^+ and C^- of patterns corresponding to a $+1$ and to a -1 output respectively.
 - 2) Add another unit to layer $\ell + 1$. Use the perceptron algorithm to train the new unit with the smallest unfaithful class obtained so far. Determine the new classes brought about by the presence of the new unit. Check whether the internal representation is faithful or not. This can be achieved by a direct inspection of classes (see Fig. 8.5).
 - 3) Repeat in 2) until the internal representation is faithful. Then layer $\ell + 1$ is achieved.
 - 4) Add a threshold unit to layer $\ell + 1$ and iterate in 1). The convergence theorem makes sure that the algorithm stops after the building of P layers at most. In practice convergence is achieved for a much smaller number of layers.

The tiling algorithm.

The learning mechanism rests upon a very simple remark:

*Two identical internal representations
cannot yield two different outputs.*

It is obvious that the reverse is not true: two different internal representations may yield the same output state. An *internal representation is faithful* if the hidden states $I^{\mu, \text{hid}} = \{\xi_j^{\mu, \text{hid}}\}$ corresponding to non-identical output states $I^{\mu, \text{out}} = \{\xi_i^{\mu, \text{out}}\}$ are non-identical.

The tiling algorithm has been devised for strictly feedforward categorizer networks (the last layer comprises a single neuron). The concatenation of independent categorizers may always be used to realize more general mappings. It must be noted that in strictly feedforward networks faithfulness must be achieved for every layer.

Let

$$I^{\mu, \text{in}} = \{\xi_k^{\mu, \text{in}}\}, \quad I^{\mu, \text{out}} = \xi^{\mu, \text{out}}, \quad \xi \in \{-1, +1\},$$

be the set of training patterns supplemented with the threshold unit $\xi_0^{\mu, \text{in}} \equiv 1$. The construction starts with the most simple network made of N_I input units and a single output unit (see Fig. 8.5). The perceptron algorithm is used to learn the wanted associations. If the learning session is successful the construction is achieved. In general learning is not successful. When the case arises the learning process is stopped as soon as the number e_1 ($e_1 \leq P$) of errors is stable. It is then possible to make two classes of patterns, a class \mathcal{C}^+ such as

$$\sigma_1(I^{\mu \in \mathcal{C}^+, \text{in}}) = \text{sign}\left(\sum_{k \in \mathcal{J}} J_{1k} \xi_k^{\mu, \text{in}}\right) = +1$$

and a class \mathcal{C}^- such as

$$\sigma_1(I^{\mu \in \mathcal{C}^-, \text{in}}) = -1.$$

Since there are errors some patterns are ill-classified. For example there exist patterns μ_0 belonging to \mathcal{C}^+ such as

$$\xi^{\mu_0 \in \mathcal{C}^+, \text{out}} = -1.$$

One now considers the output unit σ_1 as the first unit of a second layer which will be the hidden layer of a three-layer network. The set of states $\sigma_1(I^{\mu, \text{in}})$ makes a $1 \times P$ internal representation. The representation is unfaithful, since two patterns I^{μ_0} and I^{μ_1} belonging to \mathcal{C}^+ which have the same internal representation may be associated with two different output states. To create a faithful representation one adds a new unit σ_2 to the hidden layer. The training set for the new unit is reduced to the patterns belonging to one of the classes, for example to \mathcal{C}^+ . The perceptron algorithm is applied to class \mathcal{C}^+ , thus creating sub-classes \mathcal{C}^{++} and \mathcal{C}^{+-} . For example, the class \mathcal{C}^{++} comprises all patterns μ such that

$$\sigma_1(I^{\mu \in \mathcal{C}^{++}, \text{in}}) = \sigma_2(I^{\mu \in \mathcal{C}^{++}, \text{in}}) = +1.$$

As the task for the perceptron rule is easier when dealing with \mathcal{C}^+ than it was when dealing with the whole training set, a number of formerly ill-classified patterns such as I^{μ_0} and I^{μ_1} are now in two different sub-classes. The $2 \times P$ internal representation of the network is more faithful than the $1 \times P$ internal representation. If the $2 \times P$ representation is not faithful yet, a third neuron σ_3 is added to the hidden layer and the perceptron rule is applied to one of the four restricted classes \mathcal{C}^{++} ,

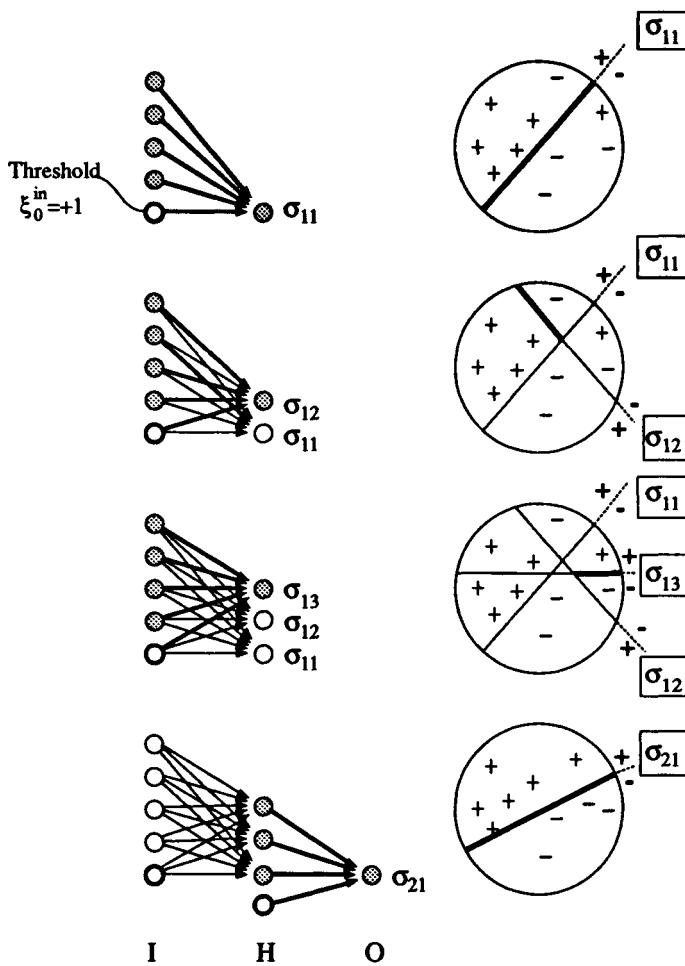


Figure 8.5. Building a feedforward network by using the tiling algorithm.

C^{+-} , C^{-+} or C^{--} that have been generated by the introduction of σ_2 . The strategy consists in choosing the unfaithful class with the smallest number of elements (of patterns). The process is repeated by adding more and more units until the $N_{\ell=2} \times P$ internal representation is faithful. As the number of patterns is finite one is certain that faithfulness can be achieved.

The perceptron algorithm is used to associate the (faithful) internal representation with the set of output patterns. In the process a threshold

unit $\sigma_0 \equiv +1$ is added to the hidden layer. Success is not guaranteed and the learning rule may not be able to do better than yielding a number e_2 of errors. The great interest of the algorithm is that there exists a convergence theorem which makes sure that

$$e_2 \leq e_1 - 1.$$

If $e_2 \neq 0$, one starts building a new hidden layer. Neurons are added until a faithful representation is achieved. A threshold unit is added and the perceptron learning rule is used to associate the internal representation displayed by the new layer with the output state $I^{\mu, \text{out}}$. The number e_ℓ of errors decreases. The whole process is repeated until one obtains $e_\ell = 0$. Then the network thus obtained categorizes correctly all the patterns of the training set.

The theorem of convergence

We assume that a faithful internal representation has already been achieved in layer ℓ . The number of units of layer ℓ is N_ℓ , with $N_\ell > 1$. A pattern, which layer ℓ does not correctly categorize, is singled out. Let I^{μ_0} be the pattern. This means that

$$\sigma_{j=1, j \in \ell}(I^{\mu_0, \text{in}}) = -\xi^{\mu_0, \text{out}}.$$

The initial connections between the first unit $\sigma_{i=1, i \in \ell+1}$ of layer $\ell+1$ and the units of layer ℓ are given the following values:

$$\begin{aligned} J_{i=1 (i \in \ell+1); j=1 (j \in \ell)} &= +1, \\ J_{i=1 (i \in \ell+1); j \in \ell, j \neq 1} &= \frac{1}{N_\ell - 1} \xi^{\mu_0, \text{out}} \sigma_j(I^{\mu_0, \text{in}}), \end{aligned} \tag{8.5}$$

where N_ℓ is the number of units in layer ℓ . The state of the first unit of layer $\ell+1$ for an input pattern $I^{\mu, \text{in}}$ is given by

$$\begin{aligned} \sigma_{i=1, i \in \ell+1}(I^{\mu, \text{in}}) &= \text{sign} \left[\sum_{j \in \ell} J_{1j} \sigma_j(I^{\mu, \text{in}}) \right] \\ &= \text{sign} \left[\sigma_{j=1, j \in \ell}(I^{\mu, \text{in}}) + \frac{\xi^{\mu_0, \text{out}}}{N_\ell - 1} \sum_{j \in \ell, j \neq 1} \sigma_j(I^{\mu_0, \text{in}}) \sigma_j(I^{\mu, \text{in}}) \right]. \end{aligned}$$

- If $\mu = \mu_0$ one finds:

$$\begin{aligned} \sigma_{i=1, i \in \ell+1}(I^{\mu_0, \text{in}}) &= \text{sign} \left[\xi^{\mu_0, \text{out}} \left(-1 + \frac{N_\ell}{N_\ell - 1} \right) \right] \\ &= \text{sign} \left[\xi^{\mu_0, \text{out}} \frac{1}{N_\ell - 1} \right] = \xi^{\mu_0, \text{out}}. \end{aligned}$$

In layer $\ell+1$ the pattern $I^{\mu_0, \text{out}}$ is well categorized.

- If $I^{\mu, \text{in}}$ is an already well-categorized pattern one has

$$\begin{aligned}\sigma_{j=1}^{i=1} (I^{\mu, \text{in}}) &= \xi^{\mu, \text{out}}, \\ \sigma_{i=1}^{j=1} (I^{\mu, \text{in}}) &= \text{sign} \left[\xi^{\mu, \text{out}} \left\{ 1 + \frac{1}{N_\ell - 1} \right. \right. \\ &\quad \times \sum_{j \in \ell, j \neq 1} \left[\xi^{\mu_0, \text{out}} \sigma_j (I^{\mu_0, \text{in}}) \right] \left[\xi^{\mu, \text{out}} \sigma_j (I^{\mu, \text{out}}) \right] \left. \right]\end{aligned}$$

The denominator in the r.h.s. of this equation involves N_ℓ terms. Its minimum value is therefore $-N_\ell$. This is a value that cannot be reached. If we assume nevertheless that it can be, then

$$\xi^{\mu_0, \text{out}} \sigma_j (I^{\mu_0, \text{in}}) = -\xi^{\mu, \text{out}} \sigma_j (I^{\mu, \text{out}})$$

for all j s including $j = 1$. If $\xi^{\mu_0, \text{out}} = -\xi^{\mu, \text{out}}$ then

$$\sigma_{j \in \ell} (I^{\mu_0, \text{in}}) = \sigma_{j \in \ell} (I^{\mu, \text{in}}),$$

which means that two patterns $I^{\mu, \text{in}}$ and $I^{\mu_0, \text{in}}$ belonging to two different classes (yielding two different outputs) have the same internal representation, which contradicts the hypothesis of faithfulness. Therefore the minimum value of the denominator is $-N_\ell + 2$ and

$$\sigma_{i=1}^{j=1} (I^{\mu, \text{in}}) = \xi^{\mu, \text{out}}.$$

Therefore the set of interactions given in Eqs (8.5) not only stabilizes an ill-categorized pattern $I^{\mu_0, \text{in}}$ but also does not perturb the stabilities of already well-categorized patterns. As a whole, the number of errors is decreased by one at least:

$$e_{\ell+1} \leq e_\ell - 1.$$

Starting from the adopted initial conditions, the perceptron algorithm can improve only the performance, and it generally yields a much smaller number of errors. One recalls that the perceptron algorithm finds a solution if a solution exists (the perceptron theorem). We just showed that at least a solution which decreases the number of errors does exist, which proves the convergence theorem of the tiling algorithm.

This algorithm has been checked by its inventors on classical problems of generalization similar to those we have seen in the preceding section. It proved also to be very efficient. It is able to find a solution for most difficult problems, that of determining a network which embeds a random Boolean function, for example. One finds that the number of units increases for the first layers, passes through a maximum and steadily decreases afterwards.

8.2.3 The Boltzmann machine

The ‘Boltzmann machine’ algorithm of Hinton and Sejnowski is an attempt to build relevant internal representations in neural networks

whose architectures are not constrained. The idea is that the probability that a pattern $I^{\mu, \text{vis}}$ will be stabilized in the *isolated network* must be equal to the probability that it occurs in the training set \mathcal{E} . An isolated network is a network whose visible units are not influenced by external stimuli. In this approach input and output patterns are treated on equal footing. Although the rationale of the algorithm is to match the internal representation of neural networks with the world of external stimuli they experience, the Boltzmann machine learning dynamics may also be viewed as a special form of associative rule. This is how it is introduced here. The link with the matching process is made clear afterwards.

To introduce the subject we consider a system with visible and hidden units, and we assume that we are given the series of P hidden states $I^{\mu, \text{hid}}$ which makes up the internal representation. This means that the pattern $I^\mu = I^{\mu, \text{vis}} \otimes I^{\mu, \text{hid}} = \{\xi_i^\mu\}$ is a fixed point of the dynamics of neural states. Let $\{J_{ij}(\mu - 1)\}$ be the set of synaptic efficacies after the step $(\mu - 1)$ of the learning dynamics has been completed. Starting from a state $I^{\mu, 0} = I^{\mu, \text{vis}} \otimes I^{\text{rand}, \text{hid}}$, where $I^{\text{rand}, \text{hid}}$ is a random state of hidden units, one looks at the state $I^{(\mu)} = \{\sigma_i^\mu\}$ one arrives at when the network relaxes freely and stops. It is likely that this state is not the desired state $I^\mu = \{\xi_i^\mu\}$. Then one may say that, if

$$\sigma_i^\mu \sigma_j^\mu = \xi_i^\mu \xi_j^\mu, \quad \text{with } \sigma_i, \xi_i \in \{+1, -1\},$$

the connection J_{ij} is correct and no modification is needed at this stage of the learning process. If, on the contrary,

$$\sigma_i^\mu \sigma_j^\mu = -\xi_i^\mu \xi_j^\mu,$$

the connection does not work in the right direction and it has to be modified according to the Hebbian rule. This remark allows us to imagine the following algorithm:

$$\Delta J_{ij}(\mu) = \varepsilon f_{ij}(\mu) \xi_i^\mu \xi_j^\mu, \quad (8.6)$$

$$\text{with } f_{ij}(\mu) = 1 - (\sigma_i^\mu \sigma_j^\mu)(\xi_i^\mu \xi_j^\mu), \quad (8.7)$$

which may be rewritten as

$$\Delta J_{ij}(\mu) = \varepsilon (\xi_i^\mu \xi_j^\mu - \sigma_i^\mu \sigma_j^\mu). \quad (8.8)$$

Extension to noisy networks is easily carried out simply by taking averages of correlation functions over the noise distribution, that is by computing $\langle \sigma_i^\mu \sigma_j^\mu \rangle$ and $\langle \xi_i^\mu \xi_j^\mu \rangle$. In Boltzmann machines noise is assumed to follow a thermal (Maxwell Boltzmann) distribution, whence the name of the algorithm (see next page).

The problem is to determine the states which appear in Eq. (8.8). Hinton and Sejnowski suggest that

- If a neuron i belongs to the set \mathcal{V} of visible units ($i \in \mathcal{V}$), the *desired state* ξ_i^μ is identical to the state of neuron i in pattern $I^{\mu, \text{vis}}$.
- If i belongs to the set \mathcal{H} of hidden units ($i \in \mathcal{H}$), the *desired state* ξ_i^μ is the relaxed state one obtains by starting from $I^{\mu,0} = I^{\mu, \text{vis}} \otimes I^{\text{random, hid}}$ and keeping the states of *visible units clamped* to $I^{\mu, \text{vis}}$.
- Whatever i ($i \in \mathcal{V}$ or $i \in \mathcal{H}$), the *relaxed state* σ_i^μ is the state one observes by starting from a configuration $I^{\mu,0}$ and letting *all units relax* afterwards. A variant consists in letting all unit relax *except the input ones which are left clamped to their input values* $\xi_i^{\mu, \text{in}}$.

In forthcoming developments the notation is made more compact by writing ‘v’ and ‘h’ instead of ‘vis’ and ‘hid’ respectively.

A starting state $I^{\mu,0}$ is defined as a state with the visible units in state $I^{\mu, \text{v}}$ and the states of hidden units chosen at random:

$$I^{\mu,0} = I^{\mu, \text{v}} \otimes I^{\text{rand, h}}$$

- 1) Start from $I^{\mu,0}$ and let the system relax while keeping $I^{\text{v}} \equiv I^{\mu, \text{v}}$ for all visible units. Compute the thermal averages $\langle \xi_i^\mu \xi_j^\mu \rangle$ for all couples $\langle ij \rangle$ (*constrained network*).
- 2) Start from $I^{\mu,0}$ and let all units free to relax or,
- 2 bis) Start from $I^{\mu,0}$ and let all units, except the input ones, free to relax. Then compute the thermal averages $\langle \sigma_i^\mu \sigma_j^\mu \rangle$ for all couples $\langle ij \rangle$ (*unconstrained network*).
- 3) Modify the efficacies according to

$$\Delta J_{ij}(\mu) = \varepsilon (\langle \xi_i^\mu \xi_j^\mu \rangle - \langle \sigma_i^\mu \sigma_j^\mu \rangle)$$

for all synaptic efficacies J_{ij} .

- 4) Iterate in 1) with a new pattern $I^{\mu', \text{v}}$.

The Boltzmann machine algorithm.

We now show that this algorithm strives to match the internal representation of the neural network, *observed on visible units*, with the training set \mathcal{E} it experiences. The set $\mathcal{E} = \{I^{\mu, \text{v}}\}$ of stimuli of visible units determines a probability distribution $\rho^{\text{E}}(I^{\text{v}})$ in the phase space of visible units. The suffix ‘E’ is for ‘environment’. For example,

$$\rho^{\text{E}}(I^{\text{v}}) = 0$$

if a given state I^v does not belong to \mathcal{E} . $\rho^E(I^v)$ is the probability distribution of visible units for the *constrained network*. On the other hand, we consider the stationary distribution of states

$$\rho(I, \infty) = \rho(I) \equiv \rho(I^v \otimes I^h)$$

for the *unconstrained network*. The distribution probability $\rho^{un}(I^v)$ of the visible units in the unconstrained network is obtained by summing out the states of hidden units:

$$\rho^{un}(I^v) = \sum_{\{I^h\}} \rho(I^v \otimes I^h). \quad (8.9)$$

The suffix ‘un’ is for ‘unconstrained networks’. We shall see that the Boltzmann machine algorithm modifies the synaptic efficacies so as to make the distribution $\rho^{un}(I^v)$ closer and closer to the distribution $\rho^E(I^v)$.

To prove this statement it is necessary to introduce the notion of *relative entropy* S^R : the entropy S^R of the system of constrained visible units relative to the system of unconstrained visible units is defined as

$$S^R = \sum_{\{I^v\}} \rho^E(I^v) \log \left(\frac{\rho^E(I^v)}{\rho^{un}(I^v)} \right). \quad (8.10)$$

This quantity is positive for all probability distributions except when the two distributions ρ^{un} and ρ^E are identical:

$$\rho^{un}(I^v) \equiv \rho^E(I^v).$$

Then the relative entropy vanishes. We show below that:

The relative entropy of a system whose synaptic efficacies are modified according to the Boltzmann machine algorithm is a non-increasing function of time.

Since the relative entropy S^R decreases, it eventually reaches a (local) minimum. If the entropy one arrives at is zero, the distribution of the states I^v of visible units of the isolated system matches that of external patterns $I^{u,v}$. This may be interpreted as saying that the algorithm has successfully built a faithful internal representation of the environment. It must be stressed, however, that in the context of Boltzmann machines the meaning of the word ‘faithful’ has, as far as the organization of internal states is concerned, never been clearly stated.

Proof of the Boltzmann machine theorem

We prove first that the relative entropy is a positive quantity. Let $\rho^a(I)$ and $\rho^b(I)$ be two probability distributions. The entropy of b relative to a is given by

$$S^R = \sum_I \rho^a(I) \log \left(\frac{\rho^a(I)}{\rho^b(I)} \right) = - \sum_I \rho^a(I) \log \left(\frac{\rho^b(I)}{\rho^a(I)} \right),$$

but one knows that

$$\log(x) \leq x - 1,$$

with $\log(x) = x - 1$, when, and only when, $x = 1$. Then:

$$\begin{aligned} -\mathcal{S}^R &\leq \sum_I \rho^a(I) \left(\frac{\rho^b(I)}{\rho^a(I)} - 1 \right) \\ &= \sum_I (\rho^b(I) - \rho^a(I)) \\ &= \sum_I \rho^b(I) - \sum_I \rho^a(I) = 1 - 1 = 0 \end{aligned}$$

and therefore $\mathcal{S}^R \geq 0$ with $\mathcal{S}^R = 0$ when, and only when, the two distributions are identical.

We now compute the derivative of the relative entropy with respect to the synaptic efficacies J_{ij} . One has

$$\frac{\partial \mathcal{S}^R}{\partial J_{ij}} = - \sum_{\{I^v\}} \frac{\rho^E(I^v)}{\rho^{un}(I^v)} \frac{\partial \rho^{un}(I^v)}{\partial J_{ij}},$$

since the distribution $\rho^E(I^v)$ is fixed and therefore does not depend on the J_{ij} s. From now on, one assumes that the asymptotic distribution of the unconstrained network is a Maxwell-Boltzmann distribution. This means that

$$\rho(I) = \frac{1}{Z} \exp -\beta H(I), \quad \text{with} \quad Z = \sum_{\{I\}} \exp -\beta H(I),$$

with $H(I) = - \sum_{\langle lm \rangle} J_{lm} \sigma_l(I) \sigma_m(I)$ and $J_{lm} = J_{ml}$. Then

$$\begin{aligned} \frac{\partial \mathcal{S}^R}{\partial J_{ij}} &= - \sum_{\{I^v\}} \frac{\rho^E(I^v)}{\rho^{un}(I^v)} \frac{\partial}{\partial J_{ij}} \left[\frac{1}{Z} \sum_{\{I^h\}} \exp \left(\beta \sum_{\langle lm \rangle} J_{lm} \sigma_l \sigma_m \right) \right] \\ \frac{\partial \mathcal{S}^R}{\partial J_{ij}} &= - \sum_{\{I^v\}} \frac{\rho^E(I^v)}{\rho^{un}(I^v)} \frac{\beta}{Z} \left[\sum_{\{I^h\}} \sigma_i \sigma_j \exp \left(\beta \sum_{\langle lm \rangle} J_{lm} \sigma_l \sigma_m \right) \right] \\ &\quad + \sum_{\{I^v\}} \frac{\rho^E(I^v)}{\rho^{un}(I^v)} \frac{\beta}{Z^2} \left[\sum_{\{I^h\}} \exp \beta \sum_{\langle lm \rangle} J_{lm} \sigma_l \sigma_m \right] \\ &\quad \times \left[\sum_{\{I \equiv I^v \otimes I^h\}} \sigma_i \sigma_j \exp \beta \sum_{\langle lm \rangle} J_{lm} \sigma_l \sigma_m \right]. \end{aligned}$$

This expression is made simple by using two equalities. On the one hand,

$$\begin{aligned} \frac{1}{Z} \sum_{\{I^h\}} \sigma_i \sigma_j \exp \left(\beta \sum_{\langle lm \rangle} J_{lm} \sigma_l \sigma_m \right) &= \sum_{\{I^h\}} \sigma_i \sigma_j \rho(I^v \otimes I^h) \\ &= \sum_{\{I^h\}} \sigma_i \sigma_j \rho(I^h | I^v) \rho^{un}(I^v) \\ &= \langle \sigma_i \sigma_j \rangle^{\text{cons}} \rho^{un}(I^v), \end{aligned}$$

where $\rho(I^h | I^v)$ is the probability of the hidden units to be in state I^h , knowing that the visible units are in states I^v . $\langle \sigma_i \sigma_j \rangle^{\text{cons}}$ is the thermal average of the correlated activities when the state of the visible units, I^v , is given.

On the other,

$$\frac{1}{Z} \sum_{\{I^h\}} \exp\left(\beta \sum_{\ell m} J_{\ell m} \sigma_\ell \sigma_m\right) = \sum_{\{I^h\}} \rho(I^v \otimes I^h) = \rho^{\text{un}}(I^v).$$

We obtain

$$\frac{\partial S^R}{\partial J_{ij}} = - \sum_{\{I^v\}} \rho^E(I^v) \beta \langle \sigma_i \sigma_j \rangle^{\text{cons}} + \sum_{\{I^v\}} \rho^E(I^v) \beta \langle \sigma_i \sigma_j \rangle^{\text{un}}.$$

Finally, since $\sum_{\{I^v\}} \rho^E(I^v) = 1$, we find

$$\frac{\partial S^R}{\partial J_{ij}} = \beta \left(\langle \sigma_i \sigma_j \rangle^{\text{un}} - \overline{\langle \sigma_i \sigma_j \rangle^{\text{cons}}} \right),$$

with

$$\overline{\langle \sigma_i \sigma_j \rangle^{\text{cons}}} = \sum_{\{I^v\}} \rho^E(I^v) \langle \sigma_i \sigma_j \rangle^{\text{cons}}.$$

If the patterns are learned one after the other this expression becomes:

$$\frac{\partial S^R(\mu)}{\partial J_{ij}} = \beta \left(\langle \sigma_i^\mu \sigma_j^\mu \rangle - \langle \xi_i^\mu \xi_j^\mu \rangle \right), \quad (8.11)$$

where our previous notation has been reintroduced. The synaptic efficacies are modified according to the rule

$$\Delta J_{ij}(\mu) = -\epsilon \frac{\partial S^R}{\partial J_{ij}} = \epsilon \left(\langle \xi_i^\mu \xi_j^\mu \rangle - \langle \sigma_i^\mu \sigma_j^\mu \rangle \right), \quad \text{with } \epsilon > 0; \quad (8.12)$$

then $\delta S^R = \sum_{ij} \frac{\partial S^R}{\partial J_{ij}} \Delta J_{ij} = -\epsilon \sum_{ij} \left(\frac{\partial S^R}{\partial J_{ij}} \right)^2 \leq 0$

and one is certain that the relative entropy decreases, which proves the theorem.

Since the relative entropy is a positive quantity the algorithm stops somewhere. If it stops at a value of zero the imprint is perfect. Otherwise it could be necessary to appeal to mechanisms which allow the synaptic dynamics to escape local minima of S^R . Thermal annealing algorithms are such mechanisms (see section 8.3.2).

The Boltzmann machine algorithm provides symmetrical connections and all the tools of statistical mechanics could be used if an analytical expression of the fixed points of the synaptic dynamics, that is to say the Hamiltonian of the system, could be available. But there is no such expression and one is reduced to observing the behavior of simulated

systems. The algorithm has been applied successfully to some simple problems, with less success to others. Beside the fact that we do not understand why a system works or does not work on a specific problem, the simulations have shown two shortcomings of the procedure:

- 1) The algorithm is very slow. It is necessary to pile up a large number of relaxation steps to obtain the N^2 correlations functions $\langle \xi_i^\mu \xi_j^\mu \rangle$ and $\langle \sigma_i^\mu \sigma_j^\mu \rangle$ which determine the modifications of the synaptic weights at every learning step. Moreover the algorithm follows a gradient dynamics with the risk of being trapped in local minima. Appealing to thermal annealing procedures stretches the convergence times still further.
- 2) Even more seriously, the rule does not work when the number of hidden units starts increasing. The syndrome is that of obsession. As learning proceeds, the basin of attraction of one of the metastable states of the network of hidden units becomes wider and wider at the expense of the basins of the other states until it occupies the whole available space. The reason is that, once the network is engaged in a basin of attraction of one of the metastable states of the hidden network, the field exerted on a hidden neuron by the visible units becomes unable to overcome the field exerted by the other hidden units at least as long as the number of visible units is too small.

From a biological point of view this situation is very unsatisfactory. The nervous central system shows many instances of large fan-in, fan-out factors. For example the number of fibers in the optical tracts of man is of the order of a million whereas the number of neurons in the primary visual cortex is of the order of a billion. As has been stated by Braitenberg, the cortex is an organ which mainly speaks to itself. The problem with the Boltzmann machine algorithm (along with other algorithms) is that the system not only speaks to itself but also learns about itself, ignoring all inputs and therefore becoming unable to respond properly to external stimuli.

How to solve this problem and how nature manages to have it solved remain open questions.

8.3 Learning in Boolean networks

8.3.1 Boolean networks

Learning is not a concept that is specific to neural networks. Any network of automata may be endowed with learning capabilities by allowing dynamical changes of the parameters of the network. From the point of view of automata theory, a neuronal network is made of units, the neurons i , which are simple automata with N_I inputs, one internal state $S_i \in \{0, 1\}$ and one output which is the internal state.

The internal state is determined by a threshold rule. The output feeds any number of units. On the other hand, Boolean networks are made of units, the Boolean gates i , which are automata with two inputs, one internal state $S_i \in \{0, 1\}$ and one output which is the internal state. There are four possible input states and the logic of the gate is defined by the string of four $(0, 1)$ -outputs corresponding to these four inputs. There are therefore $2^4 = 16$ types of Boolean gates. The internal state is determined by the logic of the gate. The output feeds any number of units (see Fig. 8.6). A few investigations have been carried out on a special family of Boolean networks, namely on Boolean arrays where the gates are placed on the vertices of a square grid. In these Boolean arrays the fan-out number is 2.

We have seen in Chapter 1 that any Boolean gate can be built by using convenient combinations of neurons and therefore neuronal networks are more universal machines than Boolean networks. Moreover it is easier to define Lyapunov or energy functions in the former networks than in the latter. Nevertheless it is interesting to consider the problem of learning in Boolean networks because the special structure of these systems and the constraints associated with the nature of gates compel one to imagine new learning paradigms.

8.3.2 Learning in feedforward Boolean networks

We consider a Boolean network made of N_I input binary units labeled $1, 2, \dots, N_I$ which can feed any number of gates taken among N_G Boolean hidden gates labeled $N_I + 1, N_I + 2, \dots, N_I + N_G$. The N_O outputs are the states of the last Boolean gates,

$$N_I + N_G - N_O + 1, N_I + N_G - N_O + 2, \dots, N_I + N_G.$$

The architecture of the network is determined by (see Fig. 8.6):

- an N_G -dimensional array Λ which fixes the types of the gates:

$$\Lambda = (t_1, t_2, \dots, t_i, \dots, t_{N_G}),$$

with $1 \leq t_i \leq 16, i = 1, \dots, N_G$.

- an $N_G \times (N_I + N_G)$ matrix \mathbf{X} which describes the connectivity of the network. This matrix has two 1's on each of its lines, one corresponding to one port of the gate whose label corresponds to the line number, and the other to the other port. All other elements are zero. In feedforward networks (which is not the case of Fig. 8.6), the upper right part of the matrix is void. The number of possible feedforward architectures corresponding to given numbers of gates N_I and N_G is

$$(16)^{N_G} [(N_G + N_I - 1)(N_G + N_I - 2) \times \dots \times (N_I)]^2.$$

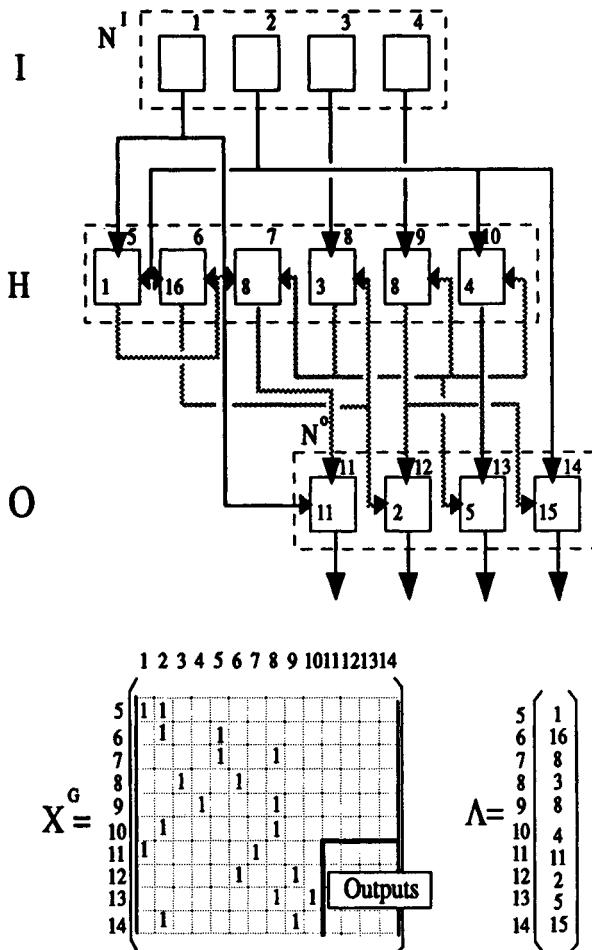


Figure 8.6. An example of a feedforward Boolean network and the two matrices X and Λ which determine its structure.

The number of possible problems, that is the number of output states for the whole set of input states, is much larger. According to section 6.1.3 it is $(2^{N_O})^{2^{N_I}}$.

The machine solves a problem if the output states are the wanted outputs for the whole set of inputs. A machine learns if it modifies its parameters so as to make the discrepancies between the wanted and the observed outputs dwindle. The parameters of the machine are embedded in the array Λ and the matrix X and therefore a learning step consists in modifying one or the other or both of these tables. As there is no clue as

to the way the tables have to be modified, the only learning procedure is by *trial and error*.

As usual the error signal is the average Hamming distance H between the observed outputs $S_i^{\mu, \text{out}} \in \{0, 1\}$ and the wanted outputs $\xi_i^{\mu, \text{out}}$. It is given by

$$H(\Lambda, \mathbf{X}) = \sum_{i=N_I+N_G-N_O+1}^{N_I+N_G} \frac{1}{P} \sum_{\mu=1}^P (S_i^{\mu, \text{out}} - \xi_i^{\mu, \text{out}})^2, \quad (8.13)$$

where P is the number of examples in the training set \mathcal{E} . The training set is a subset of the set of all instances of the problem. H is used as energy in a simulated annealing procedure.

- 1) Modify a parameter of the system at random. The result is a change ΔH of the energy which is used either
- 2a) in a Metropolis algorithm: the modification is kept with a probability

$$\bar{\omega} = \begin{cases} 1 & \text{if } \Delta H < 0, \\ \exp(-(\Delta H/T)) & \text{if } \Delta H > 0, \end{cases}$$

where T is a parameter similar to a temperature which is slowly decreased during the learning process,

- 2b) or in a Suzuki algorithm: the modification is kept with a probability

$$\bar{\omega} = \frac{1}{2}(1 - \tanh(\Delta H/T)).$$

- 3) Iterate in 1).

The simulated annealing algorithm.

One should bear in mind that both algorithms obey the detailed balance equations (see Chapter 3).

Once the learning procedure is completed it can be checked by considering two figures of merit. The first is the energy H . The other is the generalization parameter G , which is defined by

$$G(\Lambda, \mathbf{X}) = \sum_{i=N_I+N_G-N_O+1}^{N_I+N_G} \frac{1}{\mathcal{N}_I} \sum_{\mu=1}^{\mathcal{N}_I} (S_i^{\mu, \text{out}} - \xi_i^{\mu, \text{out}})^2, \quad (8.14)$$

where the summation now runs over all the $\mathcal{N}_I = 2^{N_I}$ possible instances of the problem. The maximum values of H and G are $\frac{1}{2} N_O$ when the output states are random. H is zero when all examples of the training

set have been successfully learned by the system. G is zero if the system has correctly inferred the rules of the problem, that is if it gives correct answers to instances it never learned (generalization).

The network and the learning procedure we have just described have been put forward by Patarnello and Carnevali to build an 8-bit adder. The input units are therefore made of two strings of 8 units ($N_I = 16$) and the total number of examples is $N_I = 2^{16} = 65\,536$. The output is a set of nine units which must display the result of additions. The learning procedure is restricted to modifications of the matrix \mathbf{X} . The array Λ is randomly generated and it is frozen during the learning stage. With $N_G = 160$ units, Patarnello and Carnevali found that the system learns all the examples of the training set ($H = 0$) and starts generalizing ($G = 0$) when the number P of examples is larger than $P = 160$. For $N_G = 80$ the network still learns the examples but it is unable to correctly generalize and for $N_G = 20$ it is unable to learn more than 10 examples. The learning procedure is extremely time-consuming: about 2×10^8 learning steps are necessary for the large network to achieve generalization.

There is nothing mysterious in these results. The efficiency of a learning procedure depends uniquely on the relative volume Γ of the available space of parameters which solves the problem. This is close to E. Gardner's point of view on the memorization efficiencies of neuronal networks. The difficulty here is that, for the moment at least, one does not know any technical mean of computing the value of the relative volume for Boolean networks.

8.3.3 The Aleksander model

The N_G units i of an Aleksander network have N_C binary inputs per unit, a binary internal state $S_i \in \{0, 1\}$ and one output state, identical to the internal state, which feeds N_C units. The internal state is a Boolean function f_i of the inputs:

$$S_i(\nu + 1) = f_i(\{S_{(i)}(\nu)\}), \quad (8.15)$$

where $\{S_{(i)}(\nu)\}$ is the set of units which feed the unit i . The architecture of the network is determined by the connectivity matrix \mathbf{X} and by the set of Boolean functions $\Lambda = \{f_i\}$. The Boolean networks considered by Patarnello *et al.* are special cases of Aleksander networks. The problem is to store P patterns $I^\mu = \{\xi_i^\mu\}$, in other words to determine the architecture of the network which makes the P states I^μ the fixed points of the dynamics defined by Eq. (8.15). Contrarily to the Patarnello approach, it is the set Λ of Boolean functions which is the set of learning variables. The randomly generated matrix of connectivity \mathbf{X} is frozen

during the learning stage. Every Boolean function f_i can be imagined in the form of a 2^{N_C} 1-bit memory M_i^B whose addresses are coded on N_C bits. The memory associated with unit i at address a contains the internal state 0 or 1 of that unit determined by the Boolean function f_i acting upon the set of inputs which corresponds to address a .

- 0) A 2^{N_C} three-states memory M_i^L is associated with every unit i . The three states are 0, 1 or u (undetermined). The states of the N_C inputs of the unit i determine an address in the memory. At origin time all memories of all M_i^L are in the u -state.
- 1) The patterns I^μ are presented in turn. A pattern I^μ fixes the inputs of all units of the network and therefore determines an address for every memory M_i^L . The content of these addresses are compared with the states 0 or 1 of I^μ . There are three possibilities:
 - 2a) If the state of the memory M_i^L is u , it is changed into ξ_i^μ ($= 0$ or 1).
 - 2b) If the state of the memory is $\overline{\xi_i^\mu}$ ($= 1 - \xi_i^\mu$) it becomes u .
 - 2c) If the state of the memory is identical to ξ_i^μ it is left unchanged.
- 3) The process is iterated till it hopefully becomes stationary without any memory in the u -state. Then the memories M_i^L are made identical to the memories M_i^B which determine the Boolean function $\{f_i\}$.

The Aleksander learning algorithm.

One is then certain that the memorized patterns are fixed points of the dynamics, but we ignore the size of the basins. Also it is likely that many asymptotic behaviors of the network are limit cycles, which tends to narrow the sizes of the basins. This can be avoided if one chooses a weak enough connectivity. Derrida has shown that with $1 \ll N_C < \log N_G$, the effects of loops and therefore the danger of limit cycles is avoided (see section 4.5.2). K. Wong and D. Sherrington have studied the memory storage capacity of Aleksander networks in this limit. They find that the capacity is

$$P_c = 1.08 \times \frac{2^{N_C}}{(N_C)^2}.$$

SELF-ORGANIZATION

A neural network *self-organizes* if learning proceeds without evaluating the relevance of output states. Input states are the sole data to be given and during the learning session one does not pay attention to the performance of the network. How information is embedded into the system obviously depends on the learning algorithm, but it also depends on the structure of input data and on architectural constraints.

The latter point is of paramount importance. In the first chapter we have seen that the central nervous system is highly structured, that the topologies of signals conveyed by the sensory tracts are somehow preserved in the primary areas of the cortex and that different parts of the cortex process well-defined types of information. A comprehensive theory of neural networks must account for the architecture of the networks. Up to now this has been hardly the case since one has only distinguished two types of structures, the fully connected networks and the feedforward layered systems. In reality the structures themselves are the result of the interplay between a genetically determined gross architecture (the sprouting of neuronal contacts towards defined regions of the system, for example) and the modifications of this crude design by learning and experience (the pruning of the contacts). The topology of the networks, the functional significance of their structures and the form of learning rules are therefore closely intertwined entities. There is now no global theory explaining why the structure of the CNS is the one we observe and how its different parts cooperate to produce such an efficient system, but there have been some attempts to explain at least the most simple functional organizations, those of the primary sensory areas in particular. We consider that the results are very encouraging. They proved that simple learning rules combined with simple genetically determined constraints can explain some basic cortical structures.

9.1 Self-organization in simple networks

9.1.1 Linear neural networks

Before considering more complicated systems, it is instructive to see what self-organization means in most simple networks, for example in simple linear perceptrons. This network comprises N input units j and

an unique output neuron whose average activity is σ . The response function of the output neuron is

$$\sigma = \mathcal{S}(\beta h) = \beta h$$

and the activity of the output unit that a pattern $I^\mu = \{\xi_{j \in \mathcal{I}}^\mu\}$ produces is given by

$$\sigma(I^\mu) = \beta \sum_{j=1}^N J_j \xi_j^\mu.$$

We assume that the efficacies are modified according to a symmetrical Hebbian rule

$$\Delta J_j(\mu) = \varepsilon \xi_j^\mu \sigma(I^\mu) = \varepsilon \sum_{j'}^N \xi_j^\mu \xi_{j'}^\mu J_{j'}(\mu - 1),$$

and that the parameter ε is so small that all contributions of various patterns of the learning set \mathcal{E} can be added. The modification of the vector of synaptic efficacies $\tilde{J}(\nu)$ is then given by

$$\Delta \tilde{J}(\nu) = \varepsilon \Gamma_S \cdot \tilde{J}(\nu - 1). \quad (9.1)$$

In Eq. (9.1) ν labels the number of runs, that is to say the number of times the set \mathcal{E} has been learned. Γ_S , a $N \times N$ symmetrical matrix, is the *site correlation matrix*:

$$(\Gamma_S)_{jj'} = \sum_{\mu=1}^P \xi_j^\mu \xi_{j'}^\mu.$$

Let λ_n and \tilde{v}_n be the real eigenvalues and the corresponding eigenvectors of the site correlation matrix. The solution of Eq. (9.1) is

$$\tilde{J}(\nu) = \sum_n^N \tilde{v}_n (\varepsilon \lambda_n)^\nu \tilde{v}_n^T \cdot \tilde{J}(0).$$

As learning proceeds, the behavior of the synaptic vector becomes more and more dominated by the eigenvector corresponding to the largest eigenvalue λ_1 of Γ_S :

$$\tilde{J}(\nu) \simeq (\tilde{v}_1^T \cdot \tilde{J}(0)) (\varepsilon \lambda_1)^\nu \tilde{v}_1. \quad (9.2)$$

\tilde{v}_1 is called *the principal component* of Γ_S : the result of the self-organizing process is *an extraction of the principal component of the site*

correlation matrix and the principal component is materialized in the set of synaptic efficacies. Γ_S embeds some regularities of the training set \mathcal{E} . A matrix element $(\Gamma_S)_{jj'} \neq 0$ means that, given a state ξ_j^μ of pattern I^μ on site j , the state $\xi_{j'}^\mu$ of the same pattern I^μ on site j' is not random: it depends on ξ_j^μ , with a probability which is proportional to the matrix element. The principal component is a linear combination of patterns, a ‘typical pattern’ so to speak, whose elements $(\tilde{v}_1)_j$ represent a certain average of the correlations between j and all other sites for all training patterns I^μ . Once the training is over, the activity of the output state $\sigma(I^\mu)$ yields the overlap between the input state I^μ and the principal component. It says how close the input state is to the ‘typical pattern’.

Remarks

- 1) The site correlation matrix must not be confused with the pattern correlation matrix Γ we introduced in section 7.3.4:

$$(\Gamma)_{\mu\mu'} = \sum_j^N \xi_j^\mu \xi_{j'}^{\mu'}.$$

- 2) Since a combination of linear response functions is still a linear response function, a many-layer, feedforward linear neural network is equivalent to the simple linear perceptron we have considered above. It is worth noting, however, that if the sensory signals are processed in independent pathways before they merge, every pathway extracts the principal component of its corresponding set of sensory patterns.
- 3) According to Eq. (9.2) the synaptic efficacies grow beyond any limit as learning proceeds. To avoid this inconvenience one may appeal to weak constraints:

$$\sum_j^N (J)^2 = |\tilde{J}|^2 = L.$$

The learning dynamics which is displayed in Eq. (9.1) can be viewed as a gradient dynamics determined by a cost function H^{cost} :

$$H^{\text{cost}}(\tilde{J}) = -\tilde{J} \cdot \Gamma_S \cdot \tilde{J}. \quad (9.3)$$

On the other hand the weak constraint is satisfied if the constraint term

$$H^{\text{cons}}(\tilde{J}) = (|\tilde{J}|^2 - L)^2 \quad (9.4)$$

vanishes. The constraint term modifies the learning dynamics accordingly:

$$\begin{aligned} \frac{d\tilde{J}}{dt} &= -\varepsilon \frac{d(\alpha_1 H^{\text{cons}} + \alpha_2 H^{\text{cost}})}{d\tilde{J}} \\ &= \frac{1}{\tau_1} (\Gamma_S - L \mathbf{1}) \cdot \tilde{J} - \frac{1}{\tau_2} |\tilde{J}|^2 \tilde{J}. \end{aligned} \quad (9.5)$$

This equation is known in statistical physics as a Landau Guinzburg equation. It has already been introduced in section 4.2.5, with the neural activities playing the role of synaptic efficacies. Its solution depends on the magnitude of the principal eigenvalue λ_1 relative to L . For large values of λ_1 there exists a non-trivial stable solution of synaptic efficacies, $\tilde{J} \neq 0$, allowing the principal component to be extracted. On the contrary, $\tilde{J} = 0$ is the sole stable solution for low principal eigenvalues and the property is lost.

- 4) Recursive networks that are trained by using simple Hebbian dynamics suffer from the syndrome of obsession as soon as the number of hidden units becomes an appreciable fraction of the number of input units.

9.1.2 Topological maps

We have seen in Chapter 2 that a striking feature of cortical organization is the existence of maps. There are visual maps, auditory maps and somato-sensory maps. The maps show two characteristics. On the one hand the cells respond to specific sensory patterns. For example bands of cells of the auditory cortex respond to and only to acoustic stimuli of given frequencies. On the other hand the maps materialize on the surface of the cortical sheet, topological relationships embedded in the sensory signals: cells of neighboring bands respond to tunes of neighboring frequencies. These are the two features that the learning rules proposed by Kohonen in 1981 strive to reproduce. Most models are made of two layers of neurons, a source layer \mathcal{R} (for retinal) comprising N_R units and a target layer \mathcal{C} (for cortical) comprising N_C units. The source layer may be a (very crude) model of a retina or of a cochlea and the target layer is a model of primary visual or auditory areas. It is assumed that the connections between the source and the target layer are feedforward and modifiable and that the neurons of the target layer are fully interconnected. The corresponding (intracortical) synapses are non-plastic, however.

a) Localized responses in the target layer

The first step aims at reproducing, in the spirit of grand mother cells, the specificity of responses in the target layer to a given, complex stimulus. Let us assume that we want m cells of the target layer \mathcal{C} to be active at a time. These cells may represent a cortical band for example. The synaptic efficacies between the cells of \mathcal{C} must be such that

$$\sum_{i \in \mathcal{C}}^{N_C} \sigma_i = 2m - N_C,$$

where $\sigma_i \in \{+1, -1\}$ is the state of cortical neuron i and N_C the number of units in \mathcal{C} . This is achieved by making the cost function

$$H = \left[\sum_{i \in \mathcal{C}} \sigma_i - (2m - N_C) \right]^2$$

vanish. Expanding H , using $(\sigma_i)^2 = 1$ and skipping irrelevant constants, one finds

$$H = \sum_{i, i'} \sigma_i \sigma_{i'} - 2(2m - N_C) \sum_i \sigma_i,$$

which is to be made identical to

$$H = - \sum_{\langle ii' \rangle} J_{ii'} \sigma_i \sigma_{i'} + \sum_i \theta_i \sigma_i.$$

By ignoring an irrelevant factor of 2, the identification yields

$$J_{ii'} = -1, \quad \theta_i = -J_{i0} = N_C - 2m. \quad (9.6)$$

We observe that the intra-cortical interactions are inhibitory and it is true that inhibition seems to play a prominent role in the dynamics of cortical activity. The efficacies given by Eqs (9.6) make sure that m neurons are active but a problem remains, that of packing the active units together. This may be achieved by adding local excitatory interactions $\alpha > 0$ to the efficacies $J_{ii'}$:

$$J_{ii'} = \begin{cases} -1 + \alpha & \text{if } i' \in \mathcal{V}(i), \\ -1 & \text{if } i' \notin \mathcal{V}(i). \end{cases} \quad (9.7)$$

$\mathcal{V}(i)$ is the neighborhood of i in the cortical layer \mathcal{C} . For the sake of self-consistency, $\mathcal{V}(i)$ must comprise the m nearest neighbors of i . This type of short-range excitatory and long-range inhibitory interactions is very popular in the theory of neural networks. It is called *lateral inhibition*. It is sometimes modeled by a DOG (difference of gaussians) or ‘Mexican hat’ curve (see Fig. 9.2).

b) Selecting the most responsive cortical cell

The cortical layer now displays a limited region of m neurons that shows a high activity. The problem is to locate the region. An input state $I^\mu = \{\xi_j^\mu\}_{j \in \mathcal{R}}$ determines a set of local fields $h_{i \in \mathcal{C}}^\mu$ on the neurons of the cortical layer. There are no *a priori* restrictions as regards the range of input states ξ_j^μ . It is obvious that the active region comprises the neuron i_c whose local field is the highest field. The local field is given by

$$h_{i \in \mathcal{C}}^\mu = \sum_{j \in \mathcal{R}} J_{ij} \xi_j^\mu = \tilde{J}_i \cdot \tilde{\xi}^\mu,$$

where $\tilde{\xi}^\mu = \{\xi_{j \in \mathcal{R}}^\mu\}$. The scalar product may be written as

$$h_i^\mu = \frac{1}{2} [|\tilde{J}_i|^2 + |\tilde{\xi}^\mu|^2 - (\tilde{J}_i - \tilde{\xi}^\mu)^2]. \quad (9.8)$$

One observes that if \tilde{J}_i is normalized (obeying a weak constraint condition $|\tilde{J}_i|^2 = L$), maximizing the local field h_i amounts to minimizing $(\tilde{J}_i - \tilde{\xi}^\mu)^2$, that is finding the unit i_c for which the distance between \tilde{J}_i and $\tilde{\xi}^\mu$ is minimum (whatever the units that are used to measure the lengths of these vectors):

$$|\tilde{J}_{i_c} - \tilde{\xi}^\mu| = \min_i |\tilde{J}_i - \tilde{\xi}^\mu|. \quad (9.9)$$

c) The synaptic dynamics

The synaptic efficacies $\tilde{J}_{i \in \mathcal{C}} = \{J_{i \in \mathcal{C}, j \in \mathcal{R}}\}$ are modified according to a weakly constrained Hebbian dynamics (see Eq. (7.3)).

$$\frac{d\tilde{J}_{i \in \mathcal{C}}(\mu)}{dt} = \varepsilon \left[\sigma_i(I^\mu) \tilde{\xi}^\mu - \frac{1}{\tau_i(\mu)} \tilde{J}_i \right], \quad (9.10)$$

with $\tau_i(\mu)^{-1} = \varepsilon \frac{\sigma_i(I^\mu) h_i^\mu}{L}$.

We note, in particular, that $\sigma_i = +1$ for the selected unit i_c and the m units in its neighborhood $\mathcal{V}(i_c)$ and one has

$$\frac{d\tilde{J}_i(\mu)}{dt} = \varepsilon \left[\tilde{\xi}^\mu - \frac{1}{\tau_i(\mu)} \tilde{J}_i \right], \quad i \in \mathcal{V}(i_c). \quad (9.11)$$

The equations (9.7), (9.9) and (9.10) form the basic bricks of most topological learning algorithms. In particular, they are used in the retinotopic model of Von der Malsburg (see section 9.2.1). They also make the backbone of the self-organizing algorithm of Kohonen.

9.1.3 The self-organizing model of Kohonen

a) A simple network

The self-organizing model devised by Kohonen is inspired by the mechanisms we have just described. We consider the most simple network. It is made of a single input cell ξ whose activity is restricted to a set of $P + 1$ values,

$$\xi^\mu \in \left\{ 0, \frac{1}{P}, \frac{2}{P}, \dots, \frac{\mu}{P}, \dots, 1 \right\},$$

and a one-dimensional output layer \mathcal{C} made of neurons i connected to the input unit through synapses:

$$J_{i1} \equiv J_i.$$

The synaptic efficacies are no more constrained (which would amount to saying that $J_i \equiv L = C^{\text{st}}$).

The algorithm (see below) reproduces the basic steps we introduced above. There are differences, however. For example, the selection criterion of Eq. (9.12) appears to be similar to the condition of Eq. (9.9). This is an illusion since, in the absence of a weak constraint on synaptic efficacies, J_i close to ξ^μ does not mean that i is the site which experiences the largest field created by I^μ any more. But *one keeps on considering that i_c and its neighborhood $\mathcal{V}(i_c)$ are the neurons that are made active by the input signal ξ^μ* : the neuronal dynamics has to be modified accordingly. From a physical and, all the more, from a biological point of view the meaning of criterion (9.12) is not clear. Other differences are less crucial: the relaxation parameter τ_i in Eq. (9.11) is set to $\tau_i = 1$ and the efficacies of synapses impinging on units not belonging to the neighborhood $\mathcal{V}(i_c)$ of the selected unit i_c are not modified, which is not the case in Eq. (9.11).

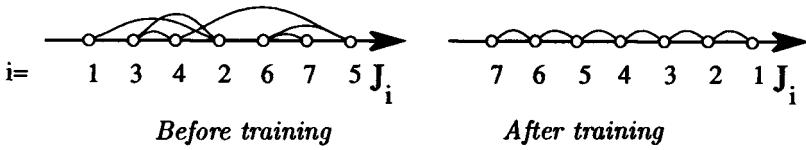
- At learning step $\nu = 0$ choose a random set of synaptic efficacies J_i .
- 1) At step ν select an activity level μ of the input unit at random
- $$\xi^\mu = \frac{\mu}{P}$$
- 2) Find the output unit i_c such as
- $$|\xi^\mu - J_{i_c}| = \min_i |\xi^\mu - J_i|. \quad (9.12)$$
- 3) Modify the synaptic efficacies according to

$$J_i(\nu + 1) = \begin{cases} J_i(\nu) + \varepsilon (\xi^\mu - J_i(\nu)) & \text{if } i \in \mathcal{V}(i_c), \\ J_i(\nu) & \text{if } i \notin \mathcal{V}(i_c). \end{cases} \quad (9.13)$$

where $\mathcal{V}(i_c)$ is a neighborhood of i_c , say its left and its right neighbor on the one-dimensional support of the output layer.

A simple Kohonen algorithm.

On the other hand the effects of the algorithm are easy to understand. The neighborhood constraint creates a topological relation between the order of input signals (which is the natural order of their amplitude $\mu = 1/P(0, 1, 2, 3, \dots, P)$) and the rank of the activated cell in the output layer. This amounts to saying that, at the end of the training, the synaptic efficacies J_i are ordered either in increasing or in decreasing order whatever the initial conditions. This is schematically depicted as follows:



On the axis of synaptic efficacies two neighboring units are linked by a curved line.

b) *The general self-organization algorithm of Kohonen*

Let us now imagine that the input layer is made of two input units whose activities are ξ_1 and ξ_2 . One assumes that the range of ξ is limited to $\xi \in [0, 1]$. Likewise, there are two connections attached to every output unit whose range is also limited to $J \in [0, 1]$. The two input activities form a two-dimensional vector $\tilde{\xi} = (\xi_1, \xi_2)$ and the two efficacies can be viewed also as a vector of a two-dimensional connection space, $\tilde{J}_i = (J_{i1}, J_{i2})$.

Therefore one has two spaces, here two squares $[0, 1]^2$, one, S^1 , with points associated with all possible activities of the input layer and the other, S^2 , with points which represent the synaptic efficacies impinging on given neuron i (see Fig. 9.1). More generally, S^1 and S^2 are spaces of dimensions N_R . The algorithm becomes the topological mapping algorithm of Kohonen.

- Start with a random set of interactions \tilde{J}_i .

 - 1) Choose a point $\tilde{\xi}^\mu$ of S^1 (a pattern I^μ of activities in the first layer).
 - 2) Find i_c of S^2 such that \tilde{J}_i is closest to $\tilde{\xi}^\mu$:
$$|\tilde{\xi}^\mu - \tilde{J}_{i_c}| = \min_i |\tilde{\xi}^\mu - \tilde{J}_i|.$$
 - 3) Modify the synaptic efficacies according to
$$\tilde{J}_i(\nu + 1) = \begin{cases} \tilde{J}_i(\nu) + \varepsilon(\nu)(\tilde{\xi}^\mu - \tilde{J}_i(\nu)) & \text{if } i \in \mathcal{V}(i_c), \\ \tilde{J}_i(\nu) & \text{if } i \notin \mathcal{V}(i_c). \end{cases}$$
 - 4) Iterate in 1).

The topological mapping algorithm of Kohonen.

In this general algorithm one still considers that the ‘most sensitive cell’ is that given by criterion (9.9) and, therefore, that an input signal $\tilde{\xi}^\mu$ triggers the activity of i_c and that of neighboring neurons $\mathcal{V}(i_c)$ of \mathcal{C} , even though this dynamics is at odds with the usual neuronal dynamics.

However, if the dimension of input space N_R increases it is harmless to place again the weak constraint on efficacies, and both neuronal dynamics, that which selects the ‘most sensitive neuron’ and that which selects the neuron experiencing the highest polarizing field, then yield the same results.

Simulations show (in Fig. 9.1) that the representative points of the synaptic efficacies tend to form a line, which reproduces the linear topology of the output units, and tends to fill all the available space in much the same way as a Peano curve does. Everything happens as if a one-dimensional space was striving to fit into a two-dimensional space. The parameter $\varepsilon(\nu)$ controls the speed of the algorithm. Experience shows that this parameter must decrease as the algorithm proceeds. This is rather similar to a thermal annealing algorithm using continuously decreasing temperatures. The conditions that the parameter has to fulfil are

$$\sum_{\nu} \varepsilon(\nu) = \infty, \quad \sum_{\nu} \varepsilon^2(\nu) < \infty.$$

For example, $\varepsilon = 1/\nu$. Up till now the space of the output layer has been one-dimensional. The very same algorithm can be used with an output layer of higher dimensionalities. This only changes the definition of the neighborhood. Let us assume, for example, that the output layer is two-dimensional. One observes that, if the set of possible inputs is a square grid, the set of synaptic efficacies becomes a similar grid. A two-dimensional space fits into another two-dimensional space and the intrinsic topology of the input signals reflects itself in the topology of the neurons which respond best to a given input. This could account for the maps, the auditory maps or the somato-sensory maps described in Chapter 2.

The Kohonen algorithm is even able to reveal more subtle relations between input items such as minimal spanning trees relating the items.

The only analytical justifications of the Kohonen algorithm were put forward by Cottrell and Fort, who proved its convergence for the most simple model, the one-dimensional model we introduced above.

9.2 Ontogenesis

9.2.1 The ontogenetic model of Von der Malsburg

We have seen that in the primary visual cortex there exist sets of neurons which respond selectively to specific stimulations. For example, sets called orientation columns respond to bar orientations. Neighboring columns respond to close orientations of the visual stimuli. The

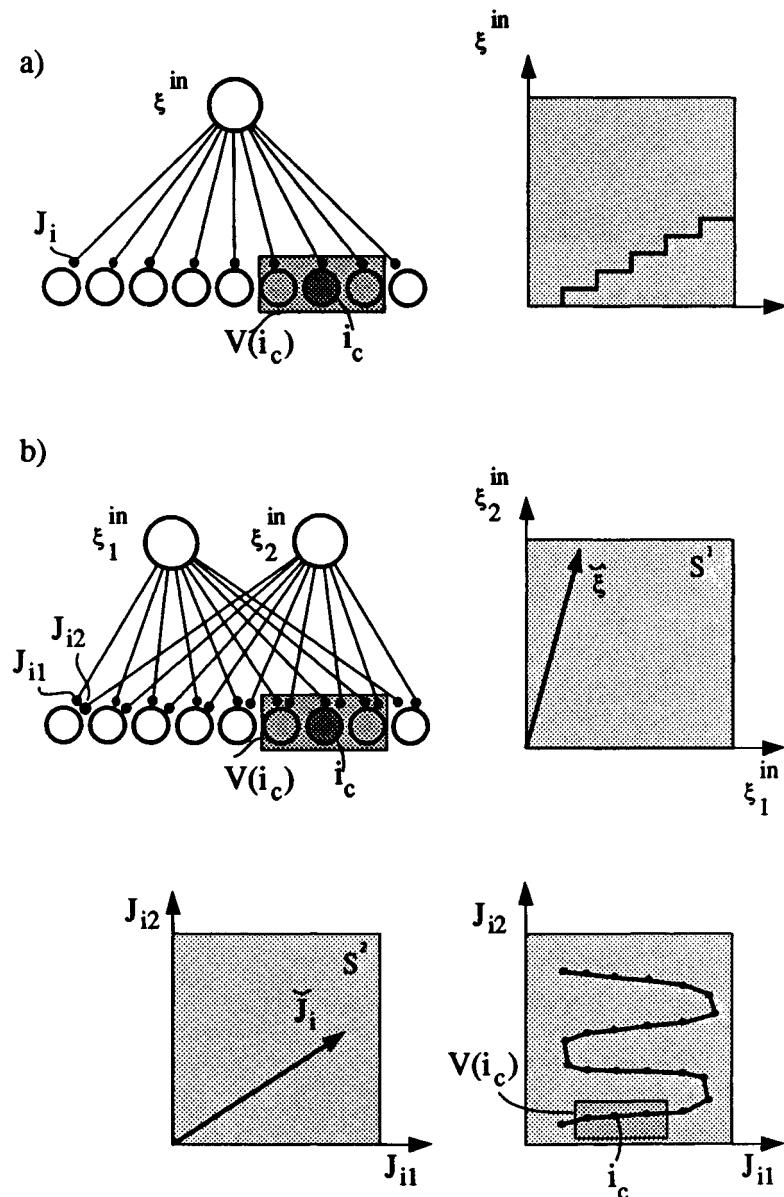


Figure 9.1. The topological learning algorithm of Kohonen.
 a) One-dimensional mapping. b) Two-dimensional mapping.
 S^1 : space of input activities, S^2 : space of interactions.
 A point of S^2 represents the interactions on a neuron i . Points corresponding to neighboring neurons are linked.

organization of columns depends heavily on learning, in particular on the visual experience in a critical period of rearing.

These results can be accounted for by a simple model proposed in 1973 by Von der Malsburg:

The structure of the network

- As in the preceding section, the basic structure, which is genetically determined (see Fig. 9.2), is made of two layers, the retina \mathcal{R} and the cortex \mathcal{C} . $\xi_{j \in \mathcal{R}}$ and $\sigma_{i \in \mathcal{C}}$ are the average activities of the retinal neurons and those of the cortical neurons respectively. They are continuous quantities.
- Each retinal neuron is connected to each cortical neuron. The synaptic efficacies $J_{i \in \mathcal{C}, j \in \mathcal{R}}$ are plastic.
- Moreover, cortico-cortical connections also do exist. The corresponding efficacies $J_{i, i' \in \mathcal{C}}$ are fixed. The interactions are of the ‘Mexican hat’ type, excitatory for the first neighbors and inhibitory for more remote neighbors: lateral inhibition phenomenon is observed in the visual cortex.

The neural and synaptic dynamics

The dynamics of the system is a coupled dynamics between

- the neuronal state dynamics,

$$\sigma_{i \in \mathcal{C}} = \mathcal{S} \left(\sum_{i' \in \mathcal{C}} J_{ii'} \sigma_{i'} + \sum_{j \in \mathcal{R}} J_{ij} \xi_j \right), \quad (9.14)$$

where the shape of the response function \mathcal{S} is depicted in Fig. 9.2, and

- a classical Hebbian synaptic dynamics,

$$\Delta J_{i \in \mathcal{C}, j \in \mathcal{R}} = \varepsilon \sigma_i \xi_j, \quad \varepsilon > 0, \quad (9.15)$$

associated with the weak constraint conditions $\sum_{j \in \mathcal{R}} |J_{ij}| = L$, a 1-norm condition.

Simulations

The stimuli are sets of retinal activities which take the shape of bars with various orientations. The stimuli are experienced by the system one after the other (Fig. 9.3). After the training is completed, one observes that packs of cortical neurons (the orientation columns) tend to be selectively active to a given stimulus. Moreover neighbor packs respond to stimuli with close orientations.

This model shows that the rather involved functional structures which physiologists observe in the cortex can be explained by the interplay of

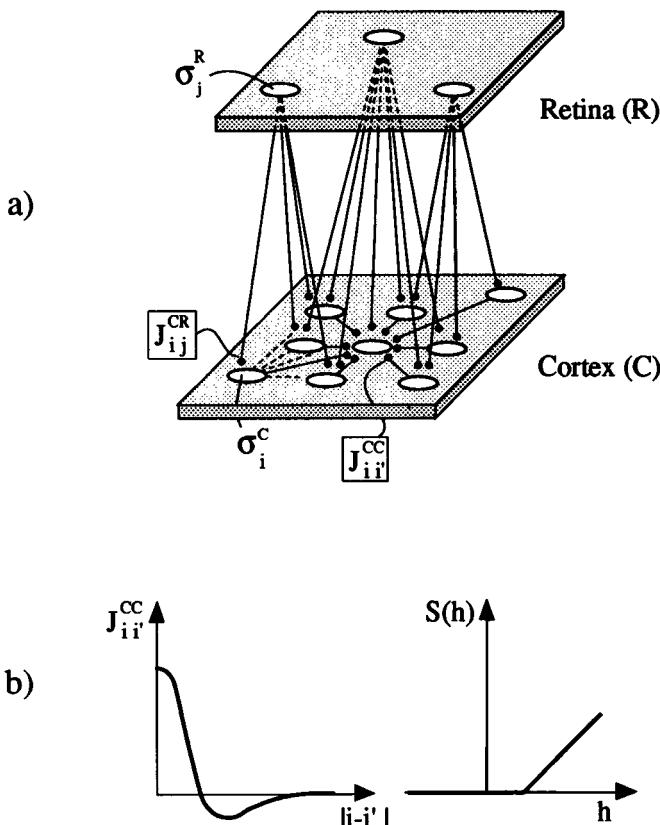


Figure 9.2. The retino-cortical structure adopted by Von der Malsburg to account for the creation of cortical regions which are specifically sensitive to bar orientation. a) The architecture of the network. b) The cortico-cortical interactions and the response function of the cortical neurons.

simple neuronal and synaptic dynamics within the framework of a simple, genetically determined design.

9.2.2 Creating feature detectors

The model of Von der Malsburg is not fully satisfactory, however, because the selectivity to bar orientations is introduced artificially by choosing very special stimuli, namely bars. With circular-shaped stimuli, for example, the cortex would have selectively responded to circles of various diameters and so on. One knows that there are cells which are sensitive to bars. Cells sensitive to circles have not been observed so far. Moreover, this sensitivity is found in newborn kittens, which indicates that

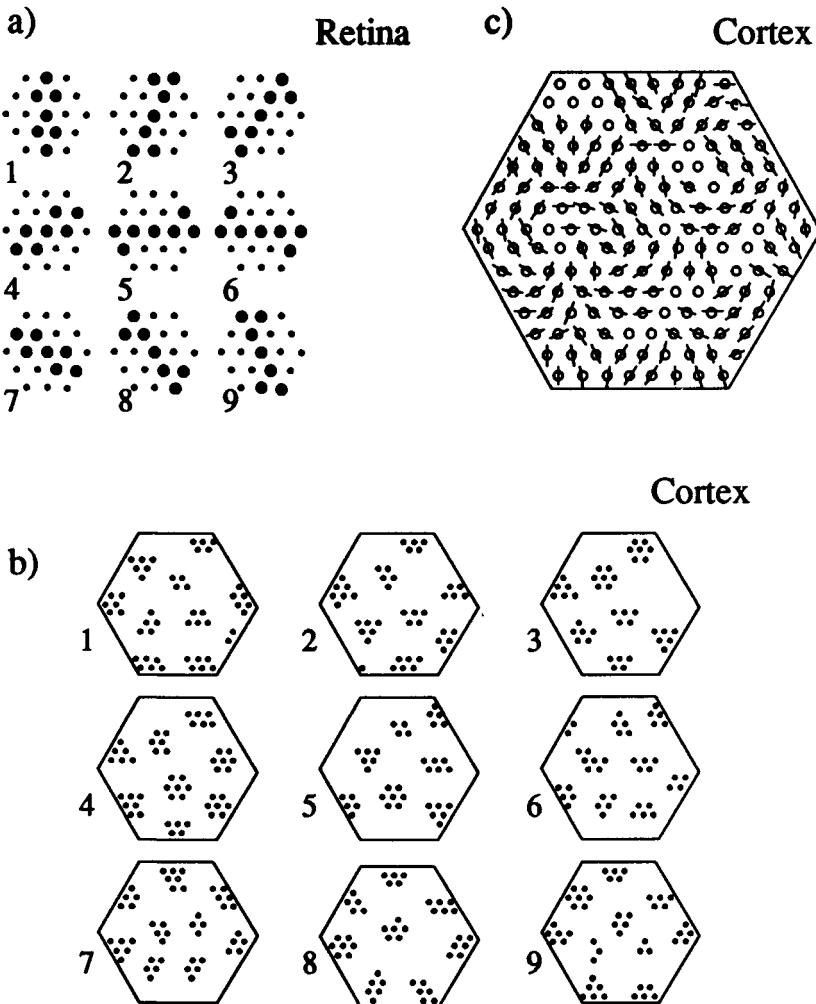


Figure 9.3. The cortical maps generated by the learning mechanism proposed by Von der Malsburg. The cortical neurons sensitive to a specific bar orientation on the retina (a) form patches of activities (b). Patches corresponding to close orientations are close to each other (c).

there exist cells displaying an orientation selectivity even when the visual experience is missing. Experience only reinforces or disables the orientation selectivity. It does not create it. Moreover, other cells are sensitive to other features such as on off cells, which respond to localized stimuli with bright centers surrounded by dark peripheries, or off on

cells, which are most sensitive to inverted stimuli.

In 1986 Linsker devised a model which accounts for the emergence of such cells.

The structure of the network

- The network is made of a number of layers labeled ℓ , with $\ell = 1, 2, \dots$ (see Fig. 9.4). The average activity of neuron $i \in \ell$ is a continuous variable $\sigma_{i\ell}$.

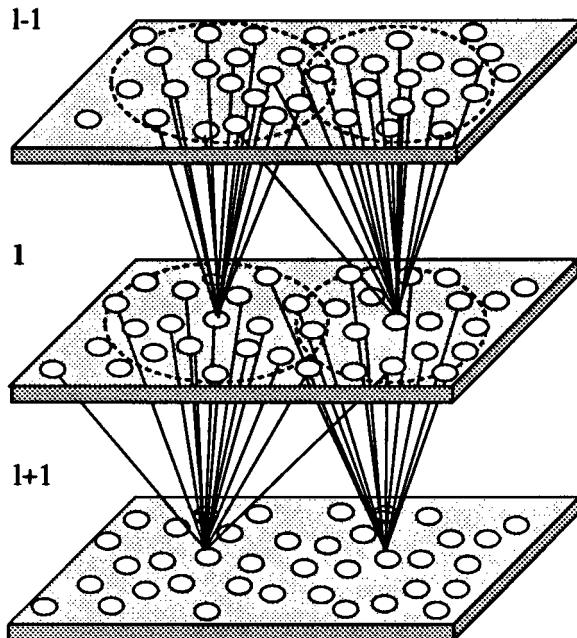


Figure 9.4. The layered structure proposed by Linsker to account for the emergence of feature detectors in the visual cortex. The distribution of neurons of layer $(\ell - 1)$ impinging on a given neuron of layer ℓ is a Gaussian which is centered on the apex of the target neuron. One can find such layered structures in the visual cortex and/or in the LGN's.

- The connections are feedforward with no intra-layer interactions. The synaptic projections are not uniformly distributed. The probability that a neuron of layer $\ell - 1$ makes a contact with a neuron of layer ℓ is maximum for the neuron of layer ℓ which lies right below it. It spreads according to a Gaussian distribution over a range r_c . These synapses are plastic.

The neuronal and synaptic dynamics

- The neuronal dynamics is a simple linear response function

$$\sigma_{i \in \ell} = \beta \sum_{j \in (\ell-1), j \in \mathcal{V}(i)} J_{ij} \sigma_j, \quad (9.16)$$

where $\mathcal{V}(i \in \ell)$ is the set of neurons $j \in \ell - 1$ connected to neuron i .

- The synaptic dynamics is Hebbian with a constant negative term which is added to account for lateral inhibition:

$$\Delta J_{i \in \ell, j \in (\ell-1)} = -D + A \sigma_i \sigma_j, \quad A, D > 0. \quad (9.17)$$

Strong irreversible constraints are used:

$$|J_{ij}| \leq L,$$

which means that the efficacies move freely according to the dynamics until the limit L is reached. Then the synaptic efficacy is frozen for ever.

Eliminating the neuronal activities, the equations of the dynamics can be transformed into the following set of equations:

$$\Delta J_{i \in \ell, j \in (\ell-1)} = -D + \sum_{j' \in \mathcal{V}(i)} Q_{jj'}^{\ell-1} \cdot J_{ij'}, \quad (9.18)$$

with

$$Q_{j \in \ell-1, j' \in \ell-1}^{\ell-1} = \beta A \sigma_j \sigma_{j'} \quad (9.19)$$

and

$$Q_{i, i' \in \ell}^{\ell} = \beta^2 \sum_{\substack{j \in \ell-1, \in \mathcal{V}(i) \\ j' \in \ell-1, \in \mathcal{V}(i')}} Q_{jj'}^{\ell-1} J_{ij} J_{i'j'}. \quad (9.20)$$

Indeed one has

$$Q_{ii'}^{\ell} = \beta A \left(\beta \sum_{\substack{j \in \ell-1 \\ \in \mathcal{V}(i)}} J_{ij} \sigma_j \right) \left(\beta \sum_{\substack{j' \in \ell-1 \\ \in \mathcal{V}(i')}} J_{i'j'} \sigma_{j'} \right).$$

Simulations

A series of random patterns is applied to the first layer $\ell = 1$, as in organisms deprived of any visual experience. The activities are decorrelated:

$$\overline{\overline{\sigma_{i \in \ell=1} \sigma_{j \in \ell=1}}} = \delta_{ij},$$

where the double bar is the symbol of time averaging. One assumes that the learning time constants are much longer than the time elapsed between the presentation of two successive patterns, so that the equations take only pattern averages into account and, according to Eq. (9.19),

$$Q_{ii' \in \ell}^{\ell} = \beta A \overline{\overline{\sigma_i \sigma_{i'}}}.$$

Then one observes that, after the training has been completed, the efficacies of synapses impinging on a given neuron of the second layer are all negative, that the synaptic efficacies impinging on a neuron of a third layer make it an on off cell and that the efficacies impinging on a neuron of the fourth layer make it a good orientation detector. The structure of deeper layers is similar to orientation columns (Fig. 9.5).

We give a qualitative explanation of these results.

Let us consider a neuron of a given layer. The set of connections it receives from the preceding layer forms a cone.

- Consider layer $\ell = 2$. Owing to the fact that the activities in layer $\ell = 1$ are uncorrelated, the Q^1 term vanishes and the term $-D$ compels all the efficacies to freeze with a negative value.
- Owing to the shape of the distribution of afferents, the Q^2 factors of a neuron of layer $\ell = 3$ are positive and are at a maximum along the axis of the cone of impinging synapses. If the parameter A is large enough the Q^2 term can be larger than the $-D$ term and the synapses in the vicinity of the axis of the cone become excitatory. The Q^2 term dwindles for synapses which are farther away from the axis. Their efficacies freeze in negative values. The cone of efficacies is therefore made of a center of positive efficacies surrounded by a periphery of negative efficacies. This distribution makes the cells of layer $\ell = 3$ most sensitive to circular spots made of bright centers and dark surroundings.
- Let us now consider a neuron $i \in \ell = 4$ of layer $\ell = 4$. The dynamics of efficacies of the synapses impinging on this particular neuron can be imagined as minimizing an energy function given by

$$H(i \in \ell) = - \sum_{jj' \in \ell-1} Q_{jj'}^{\ell-1} J_{ij} J_{ij'} + D \sum_{j \in \ell-1} J_{ij}, \quad (9.21)$$

where the Q terms are now the effective interactions between the variables J_{ij} . Owing to the structure of the preceding layers these effective interactions take the form of a Mexican hat. The interactions J modify so as to fit the ground state of a system with positive first neighbor, interactions and negative second neighbor interactions. The ground states, that is to say the states which make H as low as possible, break the symmetry of the energy function (9.21) and are made of straight lines. The synaptic efficacies tend to adopt the same configurations, with alternating bands of negative and positive values which make the neurons of this layer well suited to detect line orientations. It must be stressed that this is achieved even if the organism never saw a single straight line.

9.2.3 Learning and temporal coincidences

Most learning algorithms we have reviewed so far depend on the computation of correlated activities. For example, the Hebbian rule can be

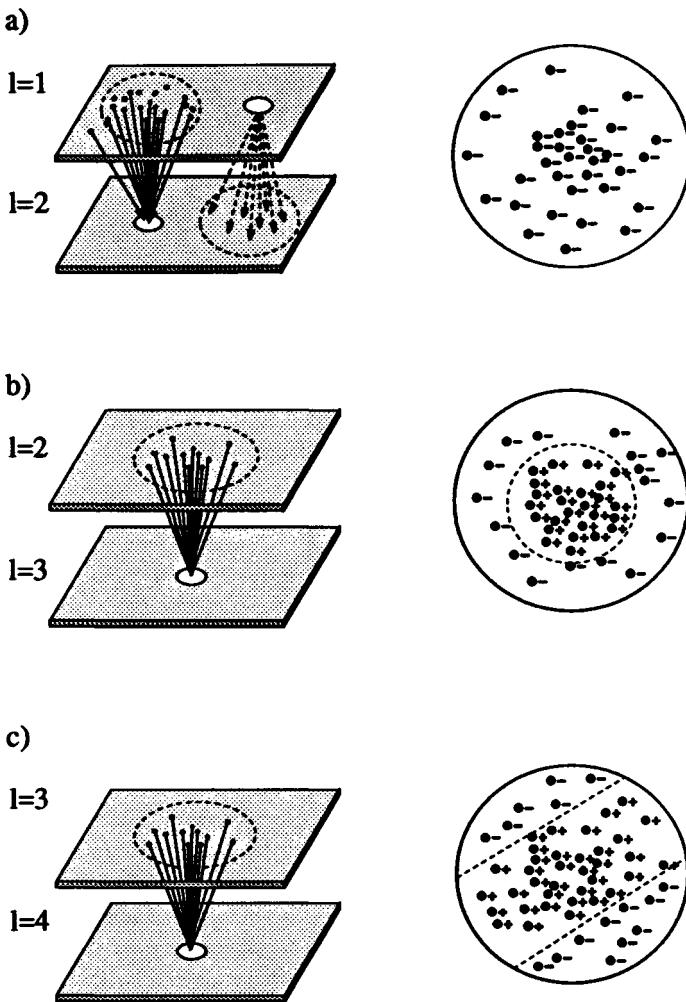


Figure 9.5. Emergence of feature detectors in the model proposed by Linsker. The model accounts for at least two types of feature detectors, the on-off cells [b) in layer $\ell = 3$] and the orientation sensitive cells [c) in layer $\ell = 4$].

written as

$$\Delta J_{ij} = \varepsilon \overline{\sigma_i} \overline{\sigma_j},$$

and this is the averaged correlation function which matters rather than the mere product of individual activities $\overline{\sigma_i} \overline{\sigma_j}$. Up to now we have not

taken the time correlation functions into account but one could wonder whether these correlations influence the synaptic efficacies through the learning algorithms. Certainly, there is much to be said on this problem and we shall not go into detail regarding the models which have been proposed. Rather, we shall only review briefly a few ideas relevant to this field.

a) *The synfire chains*

Let us consider a system of N neurons and let $z \ll N$ be the connectivity of the neurons, that is the number of neurons which are directly connected to a given neuron. There exist in general several paths to proceed from one neuron to another. The larger the number of paths between two neurons, the greater the probability that the second neuron fires if the first does. Therefore there exist groups of neurons which elicit the activities of other groups of neurons and so on, thus forming chains of activities, the synfire chains of Abeles. These firing chains have in effect been observed in the living animal. A synapse of a neuron taking part in a synfire chain often experiences simultaneous presynaptic and postsynaptic activities, since all afferents cooperate synchronously to make the neuron fire. The global effect is to reinforce the efficacies impinging on that neuron and the synfire chains must emerge as structuring entities of the cortical tissues. Little theoretical work has been carried out in that direction so far (for further discussion of this model see Chapter 12).

b) *The cocktail party problem*

During cocktail parties, there are often cross-conversations. Common experience shows that it is possible to follow the conversation we are interested in, even though the source of the relevant signal is weaker, for instance because of a larger distance, than the source of other disturbing conversations. To disentangle the signals, stereophony is important: indeed the same signal heard from a tape-recorder is incomprehensible. Von der Malsburg has proposed a model which makes use of delayed coincidences between the signals processed by the two ears to wipe out the irrelevant noise. (The reader should refer to his article for more precise information.) However, this model is probably still too crude. Once a word has been recognized, other words are anticipated by the cortex and therefore the correlation is not only between the two ears but also between the real, even distorted signal, and an expected signal. There exist other models such as that of J. Héault and Jutten, which enables a neural network to disentangle two signals starting from an unknown linear combination of the signals.

c) *Retinotopy*

During epigenesis, the afferents of the optical tract grow in the direction of the optic tectum in the cortex. It is believed that, except for the global

direction, the growing process is a stochastic process with the afferents making contacts at random in the tectum. We know however that after a while two neighboring receptors in the retina project onto two close points of the tectum.

Von der Malsburg has devised a model which can account for the organizing process. The system is made of two layers: the retina \mathcal{R} and the tectum \mathcal{C} . The retina retina connections as well as the tectum tectum connections are short range and non-plastic. The retina tectum connections are modifiable according to a Hebbian rule. Owing to the short-range interactions in the retina, the activity of a retinal receptor tends to elicit the activity of its neighbor retinal receptor. Similarly, owing to the short-range interactions in the tectum, these activities in turn tend to cause two neighbor neurons of the tectum to fire. As a whole, the structure of the system tends to activate loops of four neurons, two neighboring neurons of the retina and two neighboring neurons in the tectum. The learning process reinforces the probability that such loops fire. Retinotopy is the result of the competition between all possible loops.

We now set forth a few details of the model of A. Häussler and C. Von der Malsburg.

More on the retinotopic model

Let J_{ij} be the (positive) synaptic strength between the retinal cell j and the tectal (cortical) cell i , $i, j = 1, 2, \dots, N$. The evolution of synaptic efficacies is determined by the interplay of three different driving forces:

$$\Delta J_{ij} = \Delta J_{ij}^s + \Delta J_{ij}^c + \Delta J_{ij}^p. \quad (9.22)$$

- ΔJ_{ij}^s is a source term :

$$\Delta J_{ij}^s = a(1 - J_{ij}). \quad (9.23)$$

It is first-order in J_{ij} . The source term a creates the connections and strives to make them all equal to $J_{ij} = 1$.

• ΔJ_{ij}^c is a cooperation term. All synapses which, in the tectum, are in the vicinity of the synapse i tend to reinforce J_{ij} . Likewise, all neurons which, in the retina, are close to neuron j also tend to reinforce J_{ij} . This effect is modeled by the following equation:

$$\Delta J_{ij}^c = b J_{ij} F_{ij}, \quad (9.24)$$

$$\text{with } b > 0 \text{ and } F_{i \in \mathcal{C}, j \in \mathcal{R}} = \sum_{i' \in \mathcal{C}, j' \in \mathcal{R}} f_C(|i - i'|) f_R(|j - j'|) J_{i'j'}.$$

The functions $f_C(d)$ and $f_R(d)$ are decreasing functions of the distance d . ΔJ_{ij}^c is a second-order term. It tends to elicit bunches of large synaptic efficacies.

- ΔJ_{ij}^p is a competition term which prevents the bunches from getting too large. It is of the form

$$\Delta J_{ij}^p = -b J_{ij} G_{ij}, \quad (9.25)$$

with $G_{i \in C, j \in R} = \frac{1}{2N} \left[\sum_{j' \in R} F_{ij'} J_{ij'} + \sum_{i' \in C} F_{i'j} J_{i'j} \right]$.

The first term of ΔJ_{ij}^p limits the sum of efficacies of synapses sprouting from the axon of a given neuron i , and the second term limits the sum of efficacies impinging on the dendrites of a given neuron j . ΔJ_{ij}^p is third-order.

Equations (9.22) to (9.25) are well known from population selection theories. The solutions are of the form ‘winner takes all’; that is, only one synapse remains active for every neuron after the training stage. Moreover the topology of reinforcement is such that the target cortical neurons of two close retinal neurons are close to each other in the tectum. This is retinotopy. The process is very reminiscent of the building of maps by the Kohonen algorithm which we displayed in section 9.1.2. J.P. Changeux also appeals to this Darwinian competition cooperation mechanism to explain the pruning phenomenon of neuromuscular junctions during the development of chicken embryos.

9.2.4 Darwinian learning

A skeletal muscle is made of a number of fibers. In an adult animal the contraction of a fiber is triggered by the activity of *one motor neuron* which makes contact with the fiber at a locus called the endplate. It happens, however, that in the early stage of the development of the embryo, a given fiber is innervated by *several* neurons. This is a pruning phase, which follows the sprouting period, which reduces to one the number of synaptic contacts at the neuromuscular junction. Gouzé, Lasry and Changeux have put forward a model of population selection which accounts for these observations. They assume that the dynamics of synaptic efficacies J_i of neurons i which make contact with the fiber is autocatalytic

$$\frac{dJ_i}{dt} = \lambda(J_i)^\alpha, \quad \lambda > 0, \quad (9.26)$$

with $\alpha > 1$, and that the rate λ is proportional to the remaining amount of a certain ‘trophic factor’ which the synapses use for their development. The constant λ is therefore given by

$$\lambda = \varepsilon \left(K - \sum_{i'}^N J_{i'} \right), \quad \varepsilon, K > 0, \quad (9.27)$$

where K is the initial quantity of trophic factor. At time $t = 0$ the efficacies J_i^0 are random and small enough for λ to be considered as a

constant parameter $\lambda \simeq \varepsilon K$. Solving Eq. (9.26) then yields

$$J_i(t) = \frac{J_i^0}{[1 - \lambda(\alpha - 1)(J_i^0)^{\alpha-1} t]^{1/(\alpha-1)}}. \quad (9.28)$$

One observes that the synaptic efficacies tends to grow beyond any limit at times closer and closer to a critical time t_i^c :

$$t_i^c = \frac{1}{\lambda(\alpha - 1)(J_i^0)^{\alpha-1}}.$$

If $\alpha > 1$ the critical time depends sharply on the initial condition; the larger is the initial synaptic efficacy J_i^0 the shorter is the time t_i^c . Let us consider the neuron i^* corresponding to the largest value of J_i^0 . Since the critical time of i^* is much shorter than any other critical time, the synaptic strength of i^* strongly increases before the efficacies of other neurons have significantly evolved. This strong enhancement makes the parameter λ dwindle to zero according to Eq. (9.27). The synaptic efficacies of all neurons then stop evolving (see Eq. 9.26) in a state where the efficacy $J_{i^*}^*$ is large as compared with that of all other neurons. This is a situation where the ‘winner takes all’. Simulations with $\alpha = 3$ confirm these predictions.

9.3 Three questions about learning

The theory of learning is still in a state of infancy. Applications of the various rules to specific problems have mainly been pragmatic so far and everybody agrees that more structured approaches are needed. In particular it is necessary to make clear statements as regards what the main problems of learning are. Before closing the chapters that have been devoted to learning, we would like to give a few comments on three questions which we consider relevant in this respect.

9.3.1 The problem of coding

In the derivations we have developed in the preceding sections, we are given patterns in the form of well-defined strings of bits. These are the configurations we assume the central neural network has to deal with. The problem of coding is that of transforming raw data, for example physical sensory signals, into learnable patterns. Many coding schemes can be imagined but they are not equivalent. Depending on the coding procedure that is adopted, the very same set of raw patterns may be easy or difficult to learn. Here are some general remarks on this issue:

- a) The coding we have used so far is an arbitrary mapping between the P real patterns and P random patterns coded on a number

N of units. For example let us assume that the data are figures such as '6', '9', Random coding is achieved by associating stochastically independent strings of N bits such as (10111011) to pattern '6', (00110110) to pattern '9' and so on. Other codings have been devised. The most compact coding one can imagine is a binary coding with $N = \text{int}(\log_2 P)$, which yields (0110) for the '6', (1001) for the '9', etc. The most dilute (or sparse) coding is unary coding, where $N = P$. Then '6' is represented by (0000001000000000), '9' by (0000000001000000), This is, in actual fact, a grand mother cell representation, since a given input pattern does not elicit any activity in the output units but one. When the data shows some apparent structure, it is a good idea to have it implemented in the coding procedure. For example, an item may be characterized by a number of features:

$$I^\mu \equiv I^{\iota, \varsigma, \kappa}, \quad \mu = 1, \dots, P,$$

where ι , ς , κ label the various features. The visible units can be specialized, some for feature ι , some for features ς or κ , so that

$$I^\mu = I^\iota \otimes I^\varsigma \otimes I^\kappa.$$

For example the pattern '769' may be coded as '7' \otimes '6' \otimes '9'.

b) The very dynamics of neuronal activity makes the Hamming distance the natural distance between two neuronal states. If the data are structured according to some intrinsic distance of any type, it is desirable that this distance reflects itself in the way the data are coded: close items must be coded by patterns showing small Hamming distances. For example, in binary coding the Hamming distance between figure '7', coded (0111), and figure '8', coded (1000), is 4 whereas it is only 1 between '6', coded (0110), and '7'. In unary coding the Hamming distance is 2 whatever the figure. This is better than binary coding but still not fully satisfactory. There exists a coding procedure, needing P units as in unary coding, which respects all distances: this is the so-called 'thermometer coding' which is made explicit by saying that '6' is coded (1111110000000000) and '9' is coded (11111111000000). Their Hamming distance is 3. Let us also mention a code, the code of Gray, which has the compactness of the binary code and which respects the natural metrics of figures. This shows how important it is to define distances between raw data. In particular it is necessary to introduce distances between features which display no apparent metrics. An answer to the difficult problem of defining a convenient metric is that there do not exist metrics intrinsic to the data but rather the metrics are determined by the way the sensory tracts process input signals.

c) One may therefore wonder what sorts of coding are used by natural systems. There seems to be no general rule except that random coding is probably never used. For example coding by the olfactory bulb is unary: every molecule has its own neural receptor which is connected to the cortex by short pathways. There are many types of receptors, maybe as many as a hundred thousand. This sort of signal processing has sometimes been considered as a primitive way of coding information. This point of view has been supported by the fact that olfaction is a sense that is poorly developed in man. It is then rather surprising to observe that the coding of taste, a sense which is closely related to olfaction, seems on the contrary to be compact: there are only four types of taste buds, one for saltiness, one for bitterness, one for sweetness and one for acidity and the subtle combination of signals emitted by these four types is enough to create a whole wealth of various gustatory experiences. Color vision is similar to taste: a very limited number of types of detectors, in reality three types of retinal cones sensitive to three different ranges of light wavelengths, are enough to generate not only all natural colors, i.e. those to be seen in rainbows, but also other colors that probably do not exist outside our brains, such as pink or brown and so on. The coding of hearing seems to lie in between these two extreme cases. Neurons in the cochlea are sensitive to rather narrow ranges of sound frequencies. But there are no more than a few thousand types of cochlear neurons and it is likely that sound, like color, is somewhat recreated by the brain.

9.3.2 The problem of generalization

A system is endowed with generalization capabilities if it yields convenient responses to input patterns not belonging to the training set. One may be skeptical about the very concept of generalization. Let $\{\xi^{\mu, \text{out}}\}$ be a set of binary states which we want to consider as output states of an unknown automata network experiencing a given set of input states $\xi^{\mu, \text{in}}$, $\mu = 1, \dots, P$. The sets obey a generating rule and the question is to decide whether the learning protocol has been so successful that the trained network is not only able to reproduce the associations of the training set \mathcal{E}^t but also to produce all associations \mathcal{E}^r that the rule may generate. The point is that since there is an infinite number of machines which can realize the associations of the training set, it is meaningless to try to forecast what the outcome triggered by an extra input pattern should be. It is even impossible to decide if a finite set has been generated by a deterministic network or if it is random.

We now assume that the training set has been generated by a well-defined network and that some, but not all, information is known on this network, let us say its type (a feedforward Boolean network for

example), the number of its units, etc. If the knowledge of training set \mathcal{E}^t is enough to fully determine the network, its connectivity, the types of the gates, then the response of the network to all input patterns $\mu' > \mu$ is determined and one may say that generalization has been achieved. This leads to a first series of as yet unanswered questions on generalization: First, given an automata network and a training set, find the minimal amount of information on the network that must be kept such that the knowledge of the set is enough for a complete reconstruction of the system. Secondly, the dual problem is, for a given amount of information on the network, to find the minimal training set which would be enough to fully determine the network.

Formally, the training set \mathcal{E}^t is a subset of the set \mathcal{E}^r of all instances that the rule may create: $\mathcal{E}^t \subset \mathcal{E}^r$, and the generating rule has been extracted by learning if examples taken into \mathcal{E}^r not belonging to \mathcal{E}^t are also fixed points of the network dynamics.

Let Γ^r be the volume of the phase space of parameters, if such a volume does exist, which satisfy all possible associations \mathcal{E}^r that the rule may generate and let Γ be the volume of the phase space itself. The exhaustive study by Patarnello and Carnevali of the learning of 8-bit additions in Boolean networks (section 8.3.2) shows that generalization is achieved when

$$\frac{\Gamma^r}{\Gamma} = \frac{\mathcal{N}^t}{\mathcal{N}^r},$$

where $\mathcal{N}^t/\mathcal{N}^r$ is the number of instances of \mathcal{E}^t relative to the total number of examples generated by the rule.

In actual fact this result concerns the random search of a solution in a predetermined, large enough network. More efficient routes towards generalization may be put forward. Let Γ^t be the volume of the space of the parameters which satisfy the training set \mathcal{E}^t . One generally has

$$\Gamma^t > \Gamma^r,$$

and it is obvious that generalization is achieved if

$$\Gamma^t \simeq \Gamma^r \neq 0.$$

If $\Gamma^t \gg \Gamma^r$ the learning rule finds a set of parameters which satisfies the training set \mathcal{E}^t but responses to other patterns are incorrect (see Fig. 9.6). The system, so to speak, learns the examples ‘by heart’ and it is unable to generalize. This happens if the space of parameters is too large. In this respect it may be harmful to use sparse coding which makes the learning of \mathcal{E}^t too easy.

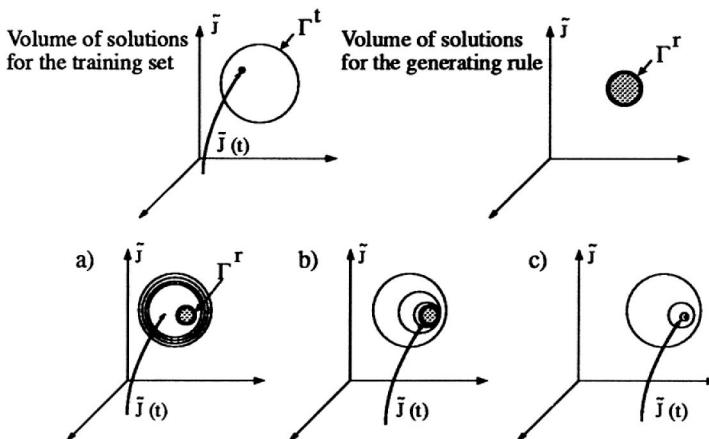


Figure 9.6. Generalization capability of a neural network.
 a) A too-large training volume: the network learns ‘by heart’.
 b) The volumes fit each other: the system can generalize.
 c) The volume which satisfies the rule vanishes: learning and therefore generalization are impossible.

If $\Gamma^r = 0$ there is no solution to the problem. This may happen if the network is too small.

The best strategy seems, therefore, to build the smallest network which may learn the training set \mathcal{E}^t . This is precisely what the constructivist algorithms such as the tiling algorithm of Mézard and Nadal (in section 8.2.2) tend to realize. One concludes that constructivist learning rules are very convenient for the sake of generalization.

These considerations are only a few hints as regards the problem of generalization. The theory still remains to be formulated.[†]

9.3.3 Learning in recursive networks with hidden units

With the exception of the Boltzmann machine all learning rules we have described so far deal either with fully connected networks or with feedforward networks. The reason is that it is not possible to control the states of hidden units in all other architectures without implementing stabilizing mechanisms which are still ill defined. Let us consider fully connected networks comprising a set \mathcal{H} of hidden units. The local field on a hidden unit is the sum of the field generated by the visible units

[†] Investigations on this problem have recently been carried out by Sara Solla *et al.*. They seem to pave the way for a genuine theory of generalization.

and of the field generated by the other hidden units:

$$h_{i \in H}^{\mu} = h_i^{\mu, \text{vis}} + h_i^{\mu, \text{hid}}.$$

Here follows a fundamental problem: either the local field h_i^{vis} outdoes the field h_i^{hid} or the reverse situation occurs.

a) The former possibility is well in line with behaviorism. The system is under the full control of the environment and there is no room for autonomous neuronal activities, be they synaptic modifications or neural states evolutions, that are not directly triggered by environmental signals. This may occur if the number of hidden units is low compared with the number of visible units. The outputs are reflex responses to input stimuli. In particular, these networks show limited self-associative memory properties.

b) In the latter case the system ‘does not mind’ about the external signals. As far as learning is concerned this is a catastrophic situation since the system is compelled to learn about itself. The result is a syndrome of obsession: a large basin of attraction develops and traps the state of the neural network, whatever signals may come from its environment. This is why the theory of learning is mainly focused on systems using architectures which avoid this sort of complication.

We have already emphasized that, according to the saying of Braitenberg, the cortex mainly speaks to itself: intracortical connections prevail over sensory cortical connexions. It is therefore necessary that the cortex is endowed with a mechanism which hinders the obsession phenomenon. Among others Rolls has suggested that one of the roles of the hippocampus could be to inhibit the memorization of already well-imprinted patterns, in the very same spirit as that of the perceptron principle (see section 7.4.2). This mechanism is probably necessary but it is not enough. It must be supported by other features of the cortical tissue such as the modularity of the cortical structures or by the fact that forward synapses, those projecting in the direction of motor areas, connect in layer IV, whereas backward synapses, projecting in the direction of sensory tracts, connect in layers II, III and V. This placing suggests that the cortical circuits could be considered as two head-to-tail feedforward networks and that, according to the situation, one network or the other could be excited. Attention could play a role in this respect.

Finally it must be emphasized that the long-term memorization mechanism is complex. The initial *supervised* imprinting process is followed by a maturation stage that may last as long as several days. In this phase the system *self-organizes*. The memorized knowledge is reprocessed to fit the already stored information. It seems necessary that any realistic model of learning and memorization should take these psychophysical observations into account.

NEUROCOMPUTATION

Neural networks are at the crossroad of several disciplines and the putative range of their applications is immense. The exploration of the possibilities is just beginning. Some domains, such as pattern recognition, which seemed particularly suited to these systems, still resist analysis. On the other hand, neural networks have proved to be a convenient tool to tackle combinatorial optimization problems, a domain to which at first sight they had no application. This indicates how difficult is the task of foreseeing the main lines of developments yet to come. All that can be done now is to give a series of examples, which we will strive to arrange in a logical order, although the link between the various topics is sometimes tenuous. Most of the applications we shall present were put forward before the fall of 1988.

10.1 Domains of applications of neural networks

Neural networks can be used in different contexts:

- For the modeling of simple biological structures whose functions are known. The study of central pattern generators is an example.
- For the modeling of higher functions of central nervous systems, in particular of those properties such as memory, attention, etc., which experimental psychology strives to quantify. Two strategies may be considered. The first consists in explaining the function of a given neural formation (as far as the function is well understood) by taking all available data on its actual structure into account. This strategy has been put forward by Marr in his theory of the cerebellum. The other strategy consists in looking for the minimal constraints that a neuronal architecture has to obey in order to account for some psychophysical property. The structure is now a consequence of the theory. If the search has been successful, it is tempting to identify the theoretical construction with biological structures which display the same organization.
- Taking things a step further, an ideal would be one where neural networks are able to solve complicated problems; that is to say, one would like to apply these systems to the domain of artificial intelligence and cognitive sciences. Significant steps have been made in this direction,

especially using neural networks as expert systems. However, applications of neural networks to other domains of AI such as solving general problems or tree-searching have still not been seriously envisioned so far.

- Finally it is possible to be completely indifferent to the biological implications of neural networks and to focus attention on their mathematical structures. Neuronal networks are systems which are well suited to tackling optimization problems. In combinatorial optimization, for example, they solve complex (NP-complete) problems quickly but approximately. They are also well adapted to ill-posed problems, those with damaged or incomplete data in particular. However, even if one is not interested in biology it can be argued that such precisely are the situations with which biological systems are daily confronted.

In this chapter a few examples of these topics are presented. They have been chosen so as to give the reader an idea of the various fields into which neural networks could be introduced. Other applications abound in literature.

10.2 Optimization

In Chapter 4 we have seen that learning is an optimization process. More generally, many information-processing mechanisms can be modeled by algorithms which strive to minimize a certain cost function. This is why we consider it convenient to start this chapter with a section devoted to optimization.

10.2.1 Mapping combinatorial optimization problems on symmetrical neural networks

We have Tank and Hopfield to thank for having realized that the range of applications of the theory of recursive neural networks is not limited to biology but that these systems could be used to solve optimization problems as well. This idea is related to the very nature of the neuronal dynamics:

$$\sigma_i(t+1) = \text{sign} \left(\sum_{j=1}^N J_{ij} \sigma_j(t) + J_{i0} \right), \quad \sigma_i \in \{-1, +1\}, \quad (10.1)$$

which compels the ‘energy’ $H(t)$,

$$H(t) = - \sum_{\langle ij \rangle} J_{ij} \sigma_i(t) \sigma_j(t) - \sum_i J_{i0} \sigma_i(t), \quad (10.2)$$

to be a non-increasing function of time provided that:

- 1) the dynamics is serial, with one neuron updated at a time;

- 2) the interactions are symmetrical, $J_{ij} = J_{ji}$;
 3) and the self-connections vanish, $J_{ii} = 0$.

In Eq. (10.2) the sum is over all couples of neurons $\langle ij \rangle$. J_{i0} ($= -\theta_i$) is the opposite of the threshold associated with neuron i .

Remark

In optimization problems the binary coding, $S_i \in \{0, 1\}$, is often more convenient than the symmetrical coding $\sigma_i \in \{-1, +1\}$ that has been used in Eqs. (10.1) and (10.2). With the change of variables,

$$\sigma_i = 2S_i - 1,$$

the energy (10.2) is written as

$$H = - \sum_{\langle ij \rangle} J_{ij} S_i S_j - \frac{1}{2} \sum_{i=1}^N \left(J_{i0} - \sum_{j=1}^N J_{ij} \right) S_i, \quad (10.3)$$

where a factor of 4 and an irrelevant constant have been skipped. The form of this expression is identical to that of Eq. (10.2):

$$H = - \sum_{\langle ij \rangle} J_{ij}^B S_i S_j - \sum_i J_{i0}^B S_i, \quad (10.4)$$

where the index B stands for ‘binary coding’. The identification of Eq. (10.4) with Eq. (10.3) yields

$$J_{ij}^B = J_{ij}, \quad J_{i0}^B = \frac{1}{2} \left(J_{i0} - \sum_{j=1}^N J_{ij} \right). \quad (10.5)$$

Conversely,

$$J_{ij} = J_{ij}^B, \quad J_{i0} = 2J_{i0}^B + \sum_{j=1}^N J_{ij}^B. \quad (10.6)$$

By using the identity

$$\mathbf{1}(x) \equiv \frac{1}{2}(\text{sign}(x) + 1), \quad (10.7)$$

one verifies that the dynamics,

$$S_i(t+1) = \mathbf{1} \left(\sum_{j=1}^N J_{ij}^B S_j(t) + J_{i0}^B \right), \quad S_i \in \{0, 1\}, \quad (10.8)$$

compels the energy (10.4) to be a non-increasing function of time.

An *optimization problem* consists of looking for a set of variables S_i , $i = 1, \dots, N$ which minimizes a cost function $f_c(\{S_i\})$ while obeying a number of constraints $f_\mu(\{S_i\})$, $\mu = 1, \dots, P$.

This sort of problem is ubiquitous in physics and it constitutes a large part of applied mathematics. The game of mapping an optimization problem onto a neural network is to build an energy landscape $H(\{S_i\})$ in the phase space of neuronal states such that the deepest minima of H are solutions to the problem. Once the mapping has been achieved one lets the system evolve according to the neuronal dynamics until a steady state is obtained. This state is a minimum of H . If, moreover, we manage to make this minimum a ground state of H , the steady state is a solution of the optimization problem (see Fig. 10.1).

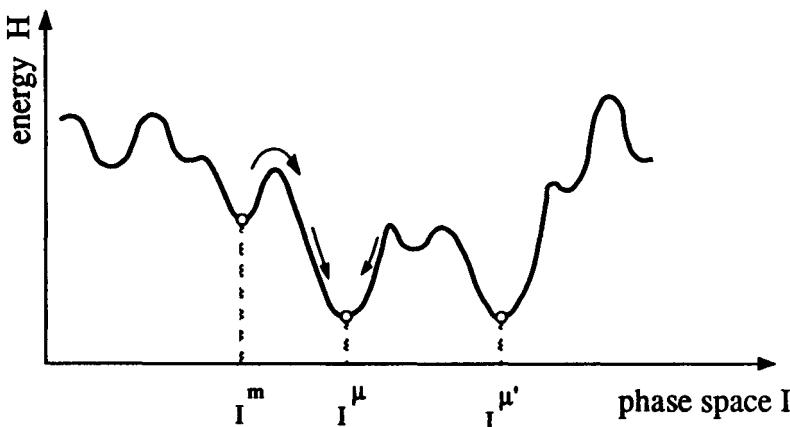


Figure 10.1. The energy landscape in the phase space of neuronal states. A smooth annealing protocol hopefully brings the system to its ground state I^μ . Noise helps to avoid the network being trapped in a metastable state such as state I^m .

The energy function H is made of a contribution arising from the cost to be minimized and contributions associated with the various constraints μ to be satisfied.

- Let us consider the cost energy H^{cost} . It is given by

$$H^{\text{cost}} = \lambda_c f_c(\{S_i\}), \quad \lambda_c > 0. \quad (10.9)$$

For Eq. (10.9) to represent an energy of type (10.4) it is necessary for f_c to be either a linear function or a quadratic function of variables S_i . More complicated functions would introduce interactions of higher orders between the neurons of the net.

- Energies H_μ^{cons} are associated with the constraints μ . One may distinguish two sorts of constraints, the strict constraints such that

$$f_\mu(\{S_i\}) = K_\mu \quad (10.10)$$

and the loose constraints such that

$$f_\mu(\{S_i\}) < K_\mu \quad \text{or} \quad f_\mu(\{S_i\}) > K_\mu. \quad (10.11)$$

Strict constraints are associated with *combinatorial optimization problems*, whereas loose constraints are associated with (*linear*) *programming problems*.

A strict constraint (10.10) is satisfied when the energy,

$$H_\mu^{\text{cons}} = \lambda_\mu \left[f_\mu(\{S_i\}) - K_\mu \right]^2, \quad \lambda_\mu > 0, \quad (10.12)$$

vanishes. The trick enables the transforming of strict constraints into energy contributions. H_μ^{cons} is a quadratic form if, and only if, f_μ is a linear function of its arguments S_i . Assuming, for example, that f_μ is quadratic, the expansion of Eq. (10.12) produces quartic contributions, such as $S_i S_j S_k S_\ell$, to energy H^{cons} . Implementing these terms into a neural network would require the introduction of four-neuron interactions.

It is not generally possible to transform loose constraints into quadratic energy functions. There exists, however, an exception which is often encountered: f_μ is an *integer* whose values are constrained by the following inequalities:

$$0 \leq f_\mu \leq 1.$$

This may be rewritten as $f_\mu = 0$ or $f_\mu = 1$. These conditions are satisfied if the energy,

$$H_\mu^{\text{cons}} = \lambda_\mu f_\mu(f_\mu - 1), \quad \lambda_\mu > 0, \quad (10.13)$$

vanishes. Otherwise H_μ^{cons} is a positive quantity.

- Finally, all contributions (10.9), (10.12) and (10.13) are lumped together,

$$H = H^{\text{cost}} + \sum_\mu H_\mu^{\text{cons}}, \quad (10.14)$$

and the interactions J_{ij} (including the thresholds J_{i0}) are determined by identifying Eq. (10.14) with (10.4).

The parameters λ_μ are Lagrange parameters. The problem is that of determining their respective values. The local field on a given unit i is

$$h_i = \sum_{\mu=1}^P \lambda_\mu \left(\sum_{j=1}^N J_{ij}^{B,\mu} S_j + J_{i0}^{B,\mu} \right).$$

If all influences on the field h_i are of equal importance, that is to say if the constraints must really be satisfied simultaneously, then the various contributions to h_i have to be of the same order of magnitude. This rule of thumb is enough to set the parameters λ_μ .

A solution of the optimization problem is obtained by appealing to the simulated annealing technique. This method, which has been imagined by Kirkpatrick and his collaborators, consists of introducing a noise term $\eta(B)$ into the state dynamics of the neural network:

$$S_i(\nu + 1) = \mathbf{1} \left[\sum_{j=1}^N J_{ij}^B S_j(\nu) + J_{i0}^B + \eta(B) \right]. \quad (10.15)$$

$\eta(B)$ is a stochastic variable whose mean value is zero and whose distribution width is B . One makes the ‘temperature’ B decrease progressively to zero according to protocols which we discuss, in the following remarks:

Some comments on the thermal annealing procedure

The energy given by Eq. (10.14) has been built so as to make its ground states the solutions of the optimization problem. But the landscape determined by H displays, in general, many other local metastable minima which eventually trap the network into undesirable states. Noise helps to escape the hanging valleys of metastable states. Indeed, when a thermodynamical noise is present the quantity which is minimal is not the average energy $\langle H \rangle$ but the free energy F (see section 4.2.1):

$$F = \langle H \rangle - BS,$$

where S is the entropy of the system. The noise is a thermodynamical noise if the transition probabilities obey the detailed balance principle (see section 3.3.2). The larger is the noise the smoother is the free energy landscape. At very high noise there is one valley. When the noise decreases this valley splits into a large valley and a narrower valley, for example. If the noise decays slowly enough the representative point of the state of the system is trapped into the larger valley. At still lower noises this valley splits in turn and the point is trapped into the widest of the newly created valleys. If according to common sense the widest valleys are the deepest ones, the point remains trapped in the deepest possible point of the free energy landscape when the noise level vanishes. As at zero noise the free energy F becomes identical to the energy H , the thermal annealing, as this slow cooling process is called, eventually traps the ground state of H .

In general the parameters of H are chosen so as to make the energy an extensive quantity, that is a quantity which is proportional to N , the size of the network. In

many optimization problems, however, the number of states is an exponential function of N . The entropy therefore scales as

$$S = \log N! \simeq N \log N,$$

which means that noise must scale as $B/\log N$.

There also remains the difficult problem of the cooling protocol. Geman *et al.* have shown that if

$$B(t) = \frac{AN}{\log(t)}$$

the system finds the solution. But this means that the time which is necessary for the neural network to find a solution increases exponentially with N . This is no surprise since a hard problem is, in actual fact, defined as a problem whose solution requires a number of steps which scales exponentially with its size N .

On the other hand one knows from statistical mechanics that the average properties of a system change most when its specific heat is large. Indeed, a large specific heat means that a small change of temperature (of noise) allows the system to explore large pools of states in the phase space. Therefore these are regions of noise in which it is good to dwell. According to this remark a well-adapted noise protocol would obey

$$\frac{dB(t)}{dt} = \epsilon \frac{d^2 F(B(t))}{dB^2}, \quad \epsilon < 0,$$

an equation which is rather hard to solve.

Finally it must be stressed that the way the state I moves in the phase space, and therefore the capacity of the annealing process to find a good, if not the best, solution, depends greatly on the dynamics. From this point of view parallel dynamics, instead of the asynchronous one-neuron dynamics we have used up to now, would certainly speed up greatly the convergence time and probably the quality of the found solution.

It is worth noting that the parallel dynamics quoted here is not equivalent to the parallel updating we introduced in Chapter 3. In the latter dynamics all states of a block of neurons are updated independently of each other according to the one-neuron flipping probability. The process can therefore be defined for any kind of one-neuron dynamics. The former, on the contrary, is meaningful only for any system whose dynamics is determined by an energy function H , that is for symmetrically connected networks. It consists of computing all the energies that a block of neurons can take by flipping one or several of the neuronal states and by choosing the block state according to the (Maxwell Boltzmann) detailed balance probabilities computed for all possible states of the block.

10.2.2 Combinatorial optimization: a few examples

The examples presented in this section are typical of a number of combinatorial optimization problems.

a) The map coloring problem

This classical problem consists of coloring the C countries of a planar map by using only four colors in such a way that no two bordering countries are given the same color. It is known that there always exists a solution to the problem (the four colors theorem).

The map can be transformed into a graph with each vertex associated with every country i . A vertex i of the graph is connected with another vertex j if the countries i and j share a common border. Therefore the elements v_{ij} of the connectivity matrix are given by

$$v_{ij} = \begin{cases} 1 & \text{if } i \text{ and } j \text{ share a common border,} \\ 0 & \text{otherwise.} \end{cases}$$

Let us now code the problem. Let α label the four colors. We define the neuronal state $S_{i\alpha}$ as

$$S_{i\alpha} = \begin{cases} 1 & \text{if country } i \text{ is given color } \alpha, \\ 0 & \text{otherwise.} \end{cases}$$

This is a unary coding. The constraint is that a country is to be painted with only one color. This constraint amounts to saying that only one neuron out of the four associated with a given country i is allowed to fire and that the energy,

$$H_i^{\text{cons}} = \lambda_0 \left(\sum_{\alpha=1}^4 S_{i\alpha} - 1 \right)^2,$$

must vanish. Using the identity $(S_{i\alpha})^2 \equiv S_{i\alpha}$, skipping an irrelevant constant and adding all contributions for the various vertices, the energy becomes

$$H^{\text{cons}} = \lambda_0 \sum_i \left[\sum_{\alpha} \sum_{\beta \neq \alpha} S_{i\alpha} S_{i\beta} - \sum_{\alpha} S_{i\alpha} \right]. \quad (10.16)$$

On the other hand, if $v_{ij} = 1$ (i and j are connected) and if the color α of i and the color β of j are the same there is a price to pay, which is

$$\delta H^{\text{cost}} = \lambda_1 v_{ij} \delta_{\alpha\beta} S_{i\alpha} S_{j\beta}.$$

Combining all contributions yields

$$H = \sum_{i,j} \sum_{\alpha,\beta} (\lambda_0 \delta_{ij} (1 - \delta_{\alpha\beta}) + \lambda_1 v_{ij} \delta_{\alpha\beta}) S_{i\alpha} S_{j\beta} - \lambda_0 \sum_{i,\alpha} S_{i\alpha}. \quad (10.17)$$

This expression is to be made identical to the standard energy,

$$H = - \sum_{(i\alpha, j\beta)} J_{i\alpha, j\beta} S_{i\alpha} S_{j\beta} - \sum_{i\alpha} J_{i\alpha, 0}^B S_{i\alpha}, \quad (10.18)$$

where $J_{i\alpha, 0}^B = -\theta_{i\alpha}^B$ is the (opposite) value of the threshold of neuron $i\alpha$. One finds

$$J_{i\alpha, j\beta} = -2(\lambda_0 \delta_{ij} (1 - \delta_{\alpha\beta}) + \lambda_1 v_{ij} \delta_{\alpha\beta}), \quad (10.19a)$$

$$J_{i\alpha, 0}^B = -\theta_{i\alpha}^B = \lambda_0, \quad (10.19b)$$

which fully determines the parameters of a neural network that may be used to solve the map coloring problem.

Remarks

- 1) The factor 2 in Eq. (10.19a) comes from the fact that the summation in Eq. (10.18) is on pairs of neurons, whereas both contributions $S_{i\alpha} S_{j\beta}$ and $S_{j\beta} S_{i\alpha}$ appear in Eq. (10.17).
- 2) It may be desirable to use the symmetrical coding $\sigma_i \in \{-1, +1\}$ rather than the binary coding $S_i \in \{0, 1\}$. Equations (10.6) yield

$$\begin{aligned} J_{i\alpha,0} &= 2J_{i\alpha,0}^B + \sum_{j\beta} J_{i\alpha,j\beta} \\ &= 2\lambda_0 - 2 \sum_{j\beta} \lambda_0 \delta_{ij} (1 - \delta_{ij}) + \lambda_1 v_{ij} \delta_{\alpha\beta} \\ &= -4\lambda_0 - 2\lambda_1 \sum_j v_{ij}. \end{aligned} \quad (10.19c)$$

- 3) The size of the network is $N = 4C$. It increases linearly with the number of countries. Simulations on 32-country problems have been successfully carried out using a 128-unit neurocomputer which we describe at the end of Chapter 11.

b) *Graph bipartition*

A graph is a set of vertices i , $i = 1, \dots, N$, and a set of links between the vertices. This set is determined by an $N \times N$ adjacency matrix v_{ij} .

$$v_{ij} = \begin{cases} 1 & \text{if there is a link between the vertices } i \text{ and } j, \\ 0 & \text{otherwise.} \end{cases}$$

The problem is that of finding a partition of the graph in two sets A and B of equal sizes $\frac{1}{2}N$ so as to make the number of links between the two sets as small as possible. This is a hard (NP-complete) problem.

This problem is easier to code using the symmetrical coding rather than the binary coding. A neuron is associated with a vertex and the state of the neuron is defined by

$$\sigma_i = \begin{cases} +1 & \text{if } i \text{ belongs to } A, \\ -1 & \text{if } i \text{ belongs to } B. \end{cases}$$

The constraint is $\sum_{i=1}^N \sigma_i = 0$ and therefore

$$H^{\text{cons}} = \left(\sum_{i=1}^N \sigma_i \right)^2.$$

With the identity $(\sigma_i)^2 = 1$, and skipping an irrelevant constant, the constraint energy becomes

$$H^{\text{cons}} = \lambda_0 \sum_{\langle ij \rangle} \sigma_i \sigma_j. \quad (10.20)$$

There is a cost if there exists a link between two vertices belonging to two different sets and a bonus if the vertices are in the same set. The cost for two vertices therefore is $-v_{ij} \sigma_i \sigma_j$ and the total cost function, here a quadratic function of the activities, is given by

$$H^{\text{cost}} = -\lambda_1 \sum_{\langle ij \rangle} v_{ij} \sigma_i \sigma_j. \quad (10.21)$$

This yields the following efficacies and threshold of the network:

$$J_{i,j} = -(\lambda_0 - \lambda_1 v_{ij}), \quad \lambda_0, \lambda_1 > 0, \quad \text{and} \quad J_{i,0} = 0.$$

The graph partition problem is therefore very similar to models of long-range Ising spin glasses. It has been studied by Mézard *et al.*, who appealed to the usual techniques of statistical mechanics.

c) Assignment problem

One wants to allocate a number N of tasks i among the same number of workers α . It costs $v_{i\alpha}$ when the worker α is given the task i . The problem is that of finding the allocation which minimizes the total cost.

A neuron is associated with a couple (i, α) . Its state is given by

$$S_{i\alpha} = \begin{cases} 1 & \text{if the worker } \alpha \text{ is given the task } i, \\ 0 & \text{otherwise.} \end{cases}$$

Let us consider an $N \times N$ matrix whose elements (i, α) are neurons $S_{i\alpha}$. A legitimate allocation, one task for one worker and one worker for one task, is represented by a matrix comprising one and only one element for each row and for each column (see Fig. 10.2). A legitimate matrix is therefore a permutation matrix and the energy associated with the constraint is

$$H^{\text{cons}} = \lambda_0 \sum_{i=1}^N \left[\left(\sum_{\alpha} S_{i\alpha} \right) - 1 \right]^2 + \sum_{\alpha=1}^N \left[\left(\sum_i S_{i\alpha} \right) - 1 \right]^2,$$

which is expanded as follows:

$$\begin{aligned} H^{\text{cons}} = & \lambda_0 \sum_i \sum_{\alpha, \beta \neq \alpha} S_{i\alpha} S_{j\beta} \\ & + \lambda_0 \sum_{\alpha} \sum_{i, j \neq i} S_{i\alpha} S_{j\beta} - 2\lambda_0 \sum_{i, \alpha} S_{i\alpha}, \end{aligned}$$

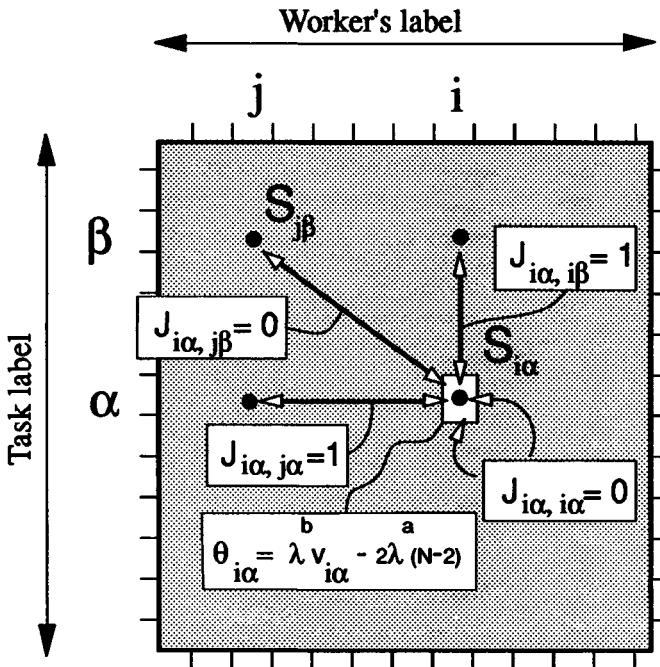


Figure 10.2. The connections and thresholds of a neuronal network devised to solve the assignment problem.

where the identity $(S_{i\alpha})^2 \equiv S_{i\alpha}$ has been taken into account. On the other hand the cost function is

$$H^{\text{cost}} = \lambda_1 \sum_{i\alpha} v_{i\alpha} S_{i\alpha}.$$

The parameters of the neural network are obtained from the identification of the energy with Eq. (10.4):

$$\begin{aligned} J_{i\alpha,j\beta} &= -2\lambda_0 [\delta_{ij} (1 - \delta_{\alpha\beta}) + \delta_{\alpha\beta} (1 - \delta_{ij})], \\ J_{i\alpha,0}^B &= 2\lambda_0 - \lambda_1 v_{i\alpha}. \end{aligned} \quad (10.22a)$$

According to Eq. (10.6) the threshold is given in the $\{-1, +1\}$ coding by

$$J_{i\alpha,0} = 4\lambda_1(N-2) - 2\lambda_0 v_{i\alpha}. \quad (10.22b)$$

It must be emphasized that the N -task problem requires a network comprising N^2 neurons.

d) *The ‘teachers and classes’ problem*

Finally we present the coding of a problem which is both more difficult and of greater practical importance, that of setting school timetables. The problem is that of allotting the various classes among the teaching hours that are available in a week. There are N_h school hours in the week. A professor p has to teach N_{pc} hours in a week to class c . The number of professors is N_p and the number of classes is N_c .

The problem is mapped onto a neural network comprising $N = N_h \times N_p \times N_c$ neurons. The coding is such that $S_{hpc} = 1$ if professor p teaches to class c during the hour h and $S_{hpc} = 0$ otherwise. The constraints are the following:

- The number of hours that p teaches to class c is N_{pc} .
- The number of professors who teach class c at time h is at most 1.
- The number of classes that are taught by a given professor p at hour h is at most 1.

They are given by:

$$\sum_h S_{hpc} = N_{pc}, \quad \sum_p S_{hpc} = 0 \text{ or } 1, \quad \sum_c S_{hpc} = 0 \text{ or } 1.$$

The constraint energies that are associated with the constraints are written as:

$$\begin{aligned} H_1^{\text{cons}} &= \lambda_1 \sum_{p,c} \left(\sum_h S_{hpc} - N_{pc} \right)^2, \\ H_2^{\text{cons}} &= \lambda_2 \sum_{h,c} \left(\sum_p S_{hpc} \right) \left(\sum_{p'} S_{hp'c} - 1 \right), \\ H_3^{\text{cons}} &= \lambda_3 \sum_{h,p} \left(\sum_c S_{hpc} \right) \left(\sum_{c'} S_{hpc'} - 1 \right). \end{aligned}$$

Once these formulae have been expanded, the identification with Eq. (10.4) yields (Fig. 10.3)

$$\begin{aligned} J_{hcp,h'c'p'} &= -2\lambda_1 \delta_{pp'} \delta_{cc'} (1 - \delta_{hh'}) \\ &\quad - 2\delta_{hh'} \left[\lambda_2 \delta_{cc'} (1 - \delta_{pp'}) + \lambda_3 \delta_{pp'} (1 - \delta_{cc'}) \right], \\ J_{hcp,0}^B &= -\lambda_1 (1 - 2N_{pc}). \end{aligned}$$

Simulations carried out by Gislen, Peterson and Söderberg have given exploitable (legal) results.

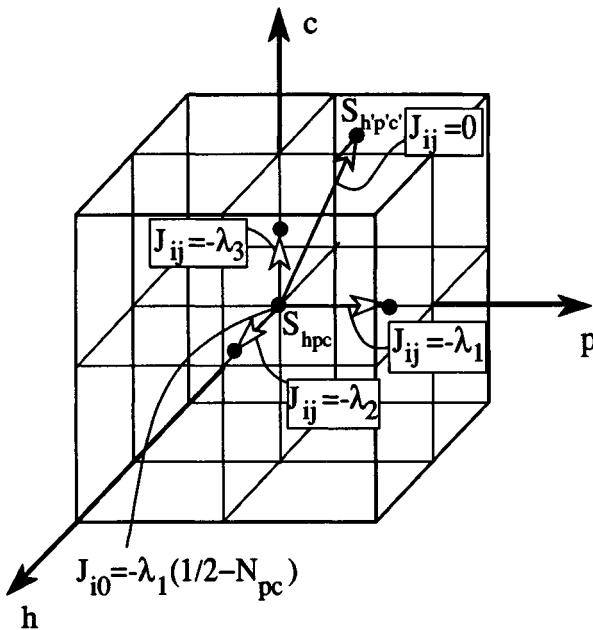


Figure 10.3. The network that solves the teachers and classes problem may be considered as a set of three-dimensional interacting units.

One may want to have an even distribution of classes all week long. The actual number of classes during hour h is

$$N_h = \sum_{c,p} S_{hcp},$$

whereas the average number of classes per hour is

$$\bar{N} = \frac{\sum_{p,c} N_{pc}}{N_h}.$$

One therefore introduces a cost which is given by

$$H^{\text{cost}} = \lambda_0 \sum_h \left(\sum_{c,p} S_{hcp} - \bar{N} \right)^2.$$

The modifications of the parameters of the network brought about by this cost function are the following:

$$\Delta J_{hcp,h'p'c'} = -2\lambda_0 \delta_{hh'}(1 - \delta_{cc'})(1 - \delta_{pp'}),$$

$$\Delta J_{hcp,0}^B = -\lambda_0(1 - 2\bar{N}).$$

For the various local fields to be of the same order of magnitude it is necessary to choose the Lagrange parameters as

$$\lambda_1 \mathcal{N}_h \simeq \lambda_2 \mathcal{N}_p \simeq \lambda_3 \mathcal{N}_c \simeq \lambda_0 \mathcal{N}_c \mathcal{N}_p.$$

10.2.3 Linear programming

Solving a linear programming problem involves finding the set of variables x_i , with $i = 1, 2, \dots, N$, which both minimizes a cost function,

$$f_c(\{x_i\}) = \sum_{i=1}^N c_i x_i, \quad (10.23)$$

a linear function of the variables, and satisfies a set of P linear inequalities,

$$f_\mu(\{x_i\}) = \sum_{i=1}^N a_{\mu i} x_i > b_\mu, \quad \mu = 1, 2, \dots, P. \quad (10.24)$$

Efficient algorithms exist, the simplex algorithm in particular, which solve the problem. In linear programming the space of possible solutions is convex and therefore the solution is unique (when it exists).

Instead of being continuous the variables could be, and often are for practical purposes, restricted to some range of integer values. The problem of linear programming with integers is much trickier than that of linear programming involving continuous variables (as explained in Fig. 10.4). In this section we first consider binary variables and the possibility of having this linear programming problem implemented in digital neural networks. The implementation of real linear programming problems in analog neural networks, which has been devised by Tank and Hopfield, is explained at the end of the section.

Up to now we have mapped optimization problems onto neural networks whose architecture is fixed, namely fully symmetrically connected networks. However, linear programming does not lend itself well to this type of network. Indeed saying that

$$f_\mu > K_\mu$$

means that *any* value of f_μ which satisfies the inequality is legal. This amounts to finding a constraint energy H^{cons} such that

$$H^{\text{cons}}(\{x_i\}) = \begin{cases} 0 & \text{if } f_\mu > K_\mu, \\ > 0 & \text{if } f_\mu \leq K_\mu. \end{cases} \quad (10.25)$$

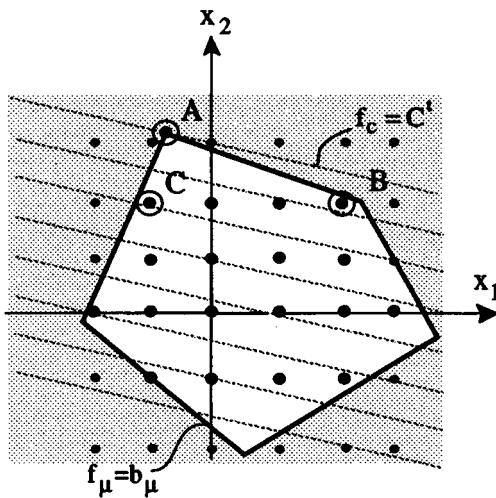


Figure 10.4. A linear programming problem in integer numbers involving two variables x_1 and x_2 and five constraints. The solution in real numbers is point A , whereas the solution in integer numbers is point B . B is not the closest point to A (this is C).

Since it is not possible to find a quadratic function of variables x_i which fulfills the conditions (10.25), the problem cannot be implemented onto symmetrically connected binary neural networks. However, if the architecture of the network is given no *a priori* restrictions there exists a solution to the problem. The only remaining condition is that the dynamics of neuronal states still follows the usual threshold dynamics.

Let us define a function $f(x)$ by

$$f(x) = \begin{cases} 0 & \text{if } x > 0, \\ |x| & \text{if } x \leq 0, \end{cases} \quad (10.26)$$

and an energy,

$$H = \lambda \sum_{i=1}^N c_i \sigma_i + \sum_{\mu=1}^P f \left(\sum_{i=1}^N a_{\mu i} \sigma_i - b_{\mu} \right), \quad \sigma_i \in \{-1, +1\}. \quad (10.27)$$

As far as the constraints are concerned, the representative point I of activities moves freely in the phase space inside the convex polyhedron defined by the inequations (10.24) and it is given some penalty if it strives to escape the polyhedron.

It is to be realized that the energy (10.27) is not a quadratic function of the activities σ_i , owing to the non-analyticity of f . A neural network must be built which will ensure that H is a non-increasing function of time ν . Let us start with an asynchronous dynamics and let us assume that the state of neuron i flips at time ν :

$$\sigma_i(\nu + 1) = -\sigma_i(\nu).$$

One computes the resulting variation of the energy H :

$$\Delta H_i(\nu) = H(\nu + 1) - H(\nu).$$

The variation of the cost energy term is

$$\Delta H_i^{\text{cost}}(\nu) = \lambda c_i (\sigma_i(\nu + 1) - \sigma_i(\nu)) = 2\lambda c_i \sigma_i(\nu + 1).$$

On the other hand, one has

$$\begin{aligned} \sum_{j=1}^N a_{\mu j} \sigma_j(\nu + 1) - b_\mu &= \sum_{j \neq i} a_{\mu j} \sigma_j(\nu + 1) + a_{\mu i} \sigma_i(\nu + 1) \\ &\quad + a_{\mu i} \sigma_i(\nu) - a_{\mu i} \sigma_i(\nu) - b_\mu, \\ &= \sum_{j \neq i} a_{\mu j} \sigma_j(\nu + 1) + a_{\mu i} \sigma_i(\nu) \\ &\quad - b_\mu + 2a_{\mu i} \sigma_i(\nu + 1) \\ &= \sum_{j=1}^N a_{\mu j} \sigma_j(\nu) - b_\mu + 2a_{\mu i} \sigma_i(\nu + 1) \end{aligned}$$

and therefore

$$\begin{aligned} \Delta H_i^{\text{cons}} &= f\left(\sum_j a_{\mu j} \sigma_j(\nu + 1) - b_\mu\right) - f\left(\sum_j a_{\mu j} \sigma_j(\nu) - b_\mu\right) \\ &= f(x + 2a_{\mu i} \sigma_i(\nu + 1)) - f(x), \end{aligned}$$

$$\text{with } x = \sum_j a_{\mu j} \sigma_j(\nu) - b_\mu.$$

One has:

- if $x > 0$ then $f(x) = 0$ and $\Delta H_i^{\text{cons}} = 0$;
- if $x < 0$ then $f(x) = -x$ and $\Delta H_i^{\text{cons}} = -2a_{\mu i} \sigma_i(\nu + 1)$.

The two formulae can be lumped together,

$$\Delta H_i^{\text{cons}} = -2a_{\mu i} \sigma_i(\nu + 1) \mathbf{1}\left(\sum_{j=1}^N -a_{\mu j} \sigma_j(\nu) + b_\mu\right);$$

and finally,

$$\Delta H_i(\nu) = 2\sigma_i(\nu + 1) \left[\lambda c_i - \sum_{\mu=1}^P a_{\mu i} \mathbf{1} \left(\sum_{j=1}^N -a_{\mu j} \sigma_j(\nu) + b_\mu \right) \right].$$

Using the identity $\mathbf{1}(x) \equiv \frac{1}{2}(1 + \text{sign}(x))$, this expression is transformed into

$$\begin{aligned} \Delta H_i(\nu) &= \sigma_i(\nu + 1) \\ &\times \left\{ 2\lambda c_i - \sum_{\mu=1}^P a_{\mu i} \left[1 + \text{sign} \left(\sum_{j=1}^N -a_{\mu j} \sigma_j(\nu) + b_\mu \right) \right] \right\}. \end{aligned} \quad (10.28)$$

One introduces new neurons τ_μ whose dynamics is given by

$$\tau_\mu(\nu + 1) = \text{sign} \left(\sum_{j=1}^N -a_{\mu j} \sigma_j(\nu) + b_\mu \right) \quad (10.29)$$

and we assume that the states of neurons i are driven by

$$\begin{aligned} \sigma_i(\nu + 1) &= \text{sign} \left(\sum_{\mu=1}^P a_{\mu i} \tau_\mu(\nu + 1) - 2\lambda c_i + \sum_{\mu=1}^P a_{\mu i} \right) \\ &= \text{sign}(X_i). \end{aligned} \quad (10.30)$$

Then the variation of the energy is

$$\Delta H_i(\nu) = -\sigma_i(\nu + 1) X_i = -X_i \text{sign}(X_i) < 0.$$

The energy H is therefore a non-increasing function of time.

The network is made of N σ -neurons and P τ -neurons. The matrix of synaptic efficacies is therefore an $(N + P) \times (N + P)$ matrix whose elements, according to the equations of the dynamics (10.29) and (10.30), are given by (see Fig. 10.5)

$$J_{k\ell} = \begin{cases} 0 & \text{if } 1 \leq k, \ell \leq N; \\ 0 & \text{if } N + 1 \leq k, \ell \leq N + P; \\ -a_{\mu i} & \text{if } k = N + \mu, \ell = i, 1 \leq \mu \leq P, 1 \leq i \leq N; \\ a_{\mu i} & \text{if } k = i, \ell = N + \mu, 1 \leq i \leq N, 1 \leq \mu \leq P. \end{cases}$$

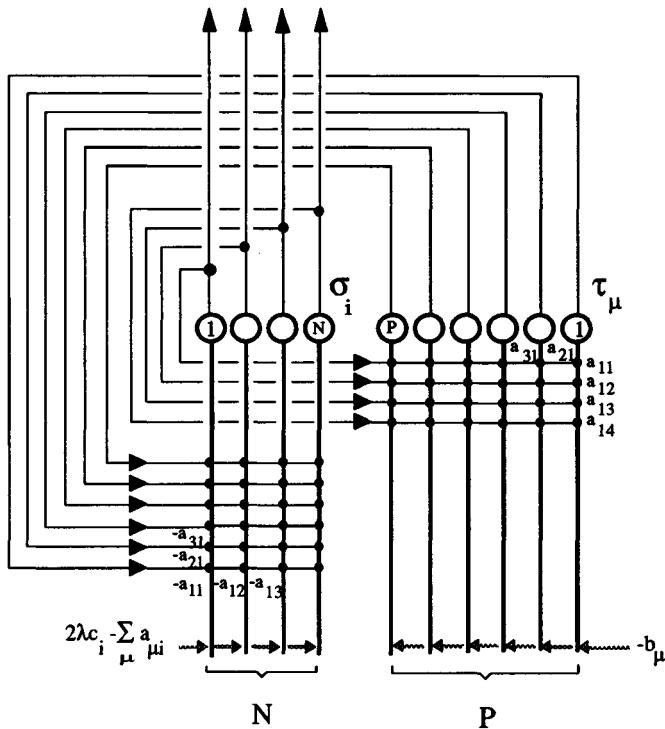


Figure 10.5. An asymmetrical architecture comprising two sorts of neurons devised to solve linear programming problems.

Likewise, the thresholds are given by

$$J_{k0} = \begin{cases} -2\lambda c_i + \sum_{\mu=1}^P a_{\mu i} & \text{if } 1 \leq k \leq N, \\ b_{\mu} & \text{if } k = N + \mu \text{ and } 1 \leq \mu \leq P. \end{cases}$$

This architecture has been *deduced* from energy considerations. Although it yields non-symmetrically connected networks we are guaranteed that the networks do not show oscillatory behavior. This is not in contradiction with the results of Chapter 3 since the energy which is invoked here is not a quadratic function of the activities. Ignoring the neurons τ , the dynamics of the neurons σ is really complicated. It is given by

$$\sigma_i(\nu + 1) = \text{sign} \left[\sum_{j=0}^P J_{i,N+\mu} \text{sign} \left(\sum_{j=0}^N J_{N+\mu,j} \sigma_j(\nu) + J_{N+\mu,0} \right) + J_{i0} \right].$$

There is also a subtlety in the derivation. Given a set of activities $\{\sigma_i\}$ of the σ -neurons, it is implicitly assumed that all states of the τ -neurons have been rearranged before the next σ -neuron can eventually flip. Therefore the dynamics cannot be random. If one scans one σ -neuron, neuron i for example, then all τ -neurons μ must be scanned before another σ -neuron i' can be updated in turn and so on. Since there is no $\sigma \sigma$ nor $\tau \tau$ connection the τ -neurons can be processed in parallel.

In actual fact the asymmetrical architecture we have derived above was first put forward by Hopfield and Tank for the implementation of linear programming problems with continuous variables. Analog networks are used whose activities are given by Eq. (3.50) (see section 3.3.5). The architecture is that depicted in Fig. 10.5 and comprises two types of analog neurons, the σ -neurons and the τ -neurons. The response curve S_σ of the σ -neurons is linear

$$S_\sigma(x) = kx,$$

whereas that S_τ of the τ -neurons is given by

$$S_\tau(x) = \begin{cases} -x & \text{if } x < 0, \\ 0 & \text{if } x > 0. \end{cases}$$

Let x_i , $i = 1, \dots, N$ be the activities of the σ -neurons and y_μ , $\mu = 1, \dots, P$ the activities of the τ -neurons. Their evolutions are given by the following equations:

$$\frac{dx_i}{dt} = -\frac{1}{T_\sigma} \left[x_i - S_\sigma \left(\sum_\mu -a_{\mu i} y_\mu - c_i \right) \right],$$

$$\frac{dy_\mu}{dt} = -\frac{1}{T_\tau} \left[y_\mu - S_\tau \left(\sum_i a_{\mu i} x_i - b_\mu \right) \right].$$

The dynamics has a fixed point $\{x_i^*, y_\mu^*\}$. For large values of k the vector $\{y_\mu^*\}$ has positive but vanishingly small components. This means that the vector defined by $\{x_i^*\}$ tends to move in the very vicinity of the surface of the limit polyhedron (but on the forbidden side of the surface). Then the dynamics strives to minimize the cost. The fixed point $\{x_i^*\}$ is the solution or close to the solution one looks for. The authors observe that the evolution does not show oscillations as long as T_σ/T_τ is small. This is explained by the special dynamics one has to use in this sort of network.

Finally it is left as an exercise to show that the function

$$H = \sum_\mu F \left(\sum_i a_{\mu i} x_i(t) - b_\mu \right) + \sum_i \left(c_i x_i(t) + \int_0^{x_i} dx S_\sigma^{-1}(x) \right),$$

where $F(x) = \int^x dx S_\tau(x)$, is a non-increasing function of time.

10.2.4 Optimization and learning algorithms

The mapping of optimization problems onto neural networks has been made possible thanks to the property of neuronal dynamics to minimize an energy function. In actual fact any dynamics which lowers a certain cost function can be used for that purpose. We have seen in Chapter 4 that most, if not all, learning rules are based upon cost minimization algorithms. It is therefore tempting to look for some adaptation of these dynamics to combinatorial optimization. This has been achieved in particular for the traveling salesman problem (TSP). The traveling salesman problem is an archetypical combinatorial optimization problem. It consists of finding the shortest tour that links all towns i of a set of towns i , $i = 1, \dots, N$ whose distances are $d_{ii'}$. A legal tour is a permutation of the towns and the number of legal tours is

$$\mathcal{N}_{\text{tour}} = \frac{1}{2N} N! = \frac{1}{2}(N-1)!,$$

since neither the origin nor the direction of the tour matter. This problem has been mapped onto symmetrically connected binary neural networks by Hopfield and Tank. We give their solution before explaining how learning algorithms may be used to solve (at least approximately) the same problem.

a) Mapping the TSP onto binary neural networks

The coding of the problem involves a N^2 -unit neural network. A neural state $S_{i\nu} = 1$ if the town i , $i = 1, \dots, N$ is visited by the salesman during day ν , $\nu = 1, \dots, N$ and $S_{i\nu} = 0$ otherwise. The constraints are that:

- Any town i is to be visited and it is to be visited once.
- The salesman must visit one and only one town during day ν .

Thus

$$\sum_{\nu=1}^N S_{i\nu} = 1 \quad \text{and} \quad \sum_{i=1}^N S_{i\nu} = 1,$$

whence the following constraint energy:

$$H^{\text{cons}} = \sum_{i=1}^N \left(\sum_{\nu=1}^N S_{i\nu} - 1 \right)^2 + \sum_{\nu=1}^N \left(\sum_{i=1}^N S_{i\nu} - 1 \right)^2. \quad (10.31)$$

On the other hand the cost of a tour is proportional to the length of the tour. There is a contribution $d_{ii'}$ to the cost if town i' is the predecessor or if it is the successor of town i . The cost energy is therefore given by

$$H^{\text{cost}} = \sum_{i\nu} \sum_{i'\nu'} d_{ii'} (\delta_{\nu,\nu'-1} + \delta_{\nu,\nu'+1}) S_{i\nu} S_{i'\nu'}. \quad (10.32)$$

Expanding Eqs (10.31) and (10.32), skipping an irrelevant factor 2 and introducing Lagrange parameters λ_1 and λ_2 yields the parameters of the network (Fig. 10.6):

$$\begin{aligned} J_{i\nu,i'\nu'} &= -\lambda_1 \left[\delta_{ii'}(1 - \delta_{\nu\nu'}) + \delta_{\nu\nu'}(1 - \delta_{ii'}) \right] \\ &\quad - \lambda_2 d_{ii'}(d_{\nu,\nu'-1} + \delta_{\nu,\nu'+1}), \\ J_{i\nu,0}^B &= \lambda_1. \end{aligned}$$

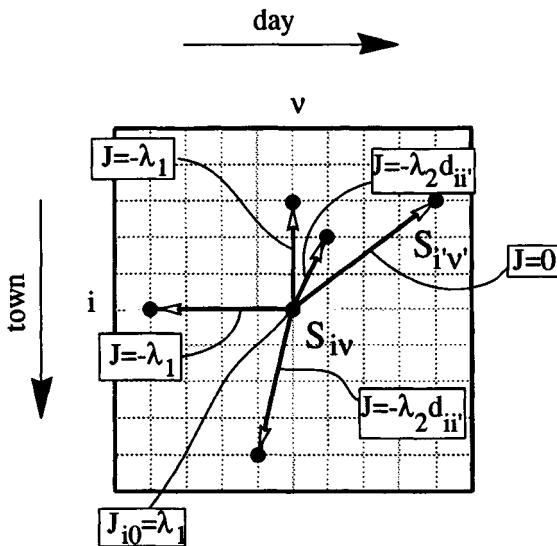


Figure 10.6. The Hopfield and Tank solution for the mapping of the TSP onto binary, symmetrically connected neural networks.

Simulations carried out on small sets of towns show that the network finds acceptable solutions, but the results compare unfavorably with those obtained by using classical algorithms such as the Lin and Kernighan algorithm (see Table 10.1).

b) The 'elastic' algorithm

The algorithm devised by Durbin and Willshaw strives to deform an initially circular circuit into the optimal tour of the traveling salesman. The circuit is made of points y_j or 'beads'. The beads are linked to each other by elastic interactions which generate a cost energy (Fig. 10.7),

$$H^{\text{cost}} = \lambda_0 \sum_j (\mathbf{y}_{j+1} - \mathbf{y}_j)^2. \quad (10.33)$$

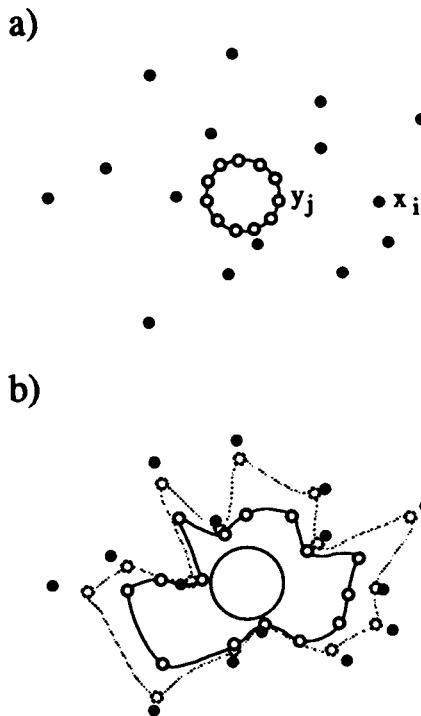


Figure 10.7. The elastic algorithm: the path (a) at time origin and (b) after its shape has been distorted by the relaxation process.

An additional constraint energy compels the circuit to go through the towns i . It is given by

$$H^{\text{cons}} = -\lambda_1 K \sum_i \log \left[\sum_j \exp \left(\frac{-(x_i - y_j)^2}{2K^2} \right) \right], \quad (10.34)$$

where x_i is the position of town i . At every step of the process the position y_j of bead j is moved along the force created by $H = H^{\text{cost}} + H^{\text{cons}}$:

$$\begin{aligned} \Delta y_j &= -K \frac{\partial H}{\partial y_j} \\ &= \lambda_1 \sum_i w_{ij}(x_i - y_j) + \lambda_0 K(y_{j+1} - 2y_j + y_{j-1}), \end{aligned} \quad (10.35)$$

with

$$w_{ij} = \frac{\exp[-(x_i - y_j)^2/2K^2]}{\sum_k \exp[-(x_i - y_k)^2/2K^2]}. \quad (10.36)$$

One makes the parameter K progressively tend to zero. The weight distribution is more and more peaked and selects the town i which is closest to point j . This is exactly the process which is brought into play in the self-organization algorithm of Kohonen which we recognize in the first term of Eq. (10.35). This algorithm gives much better results, i.e. yields shorter tours, than that of Hopfield and Tank.

c) *The Chen algorithm*

Finally we describe an algorithm put forward by Chen. A synaptic weight $J_{ii'}$ is associated with the path ii' from town i to town i' . Two tours are built by using the following greedy algorithm: one starts from town i and one chooses the next town i' of the tour according to the probability $J_{ii'}/\sum_{i''} J_{ii''}$. The towns that have been already selected are eliminated from the choice. The quality of each tour ν is characterized by a merit factor

$$\exp \left[-\lambda \frac{d(\nu)}{K} \right], \quad (10.37)$$

where $d(\nu)$ is the length of tour ν . The synaptic weights $J_{ii'}$ of links ii' which make the second tour, and these links only, are modified according to the merit factor (10.37) of tour 2 relative to tour 1:

$$J_{ii'} \mapsto J_{ii'} \exp \left[-\lambda \frac{d(2) - d(1)}{K} \right]. \quad (10.38)$$

A third tour is then generated by using the greedy algorithm in a network whose links, except the few weights that have been modified by Eq. (3.38), are the same as in the preceding step. Generally, at step $\nu + 1$ the weights of links ii' which make the tour $\nu + 1$ are modified according to

$$J_{ii'} \mapsto J_{ii'} \exp \left[-\lambda \frac{d(\nu + 1) - d(\nu)}{K} \right].$$

The algorithm stops when all weights vanish except those corresponding to the optimal tour. This is a clever algorithm since a whole set of weighted solutions are examined at every time step, with the selected solution progressively picked out over other possible tours. It is also very interesting to note that there is no need to define a cooling protocol, a sensitive point in the thermal annealing procedure. In the Chen algorithm the temperature adjusts itself as the algorithm proceeds.

d) *Comparison of performances*

The quality of normalized solutions obtained by using various algorithms on 50-town problems is displayed in Table 10.1.

Hopfield <i>et al.</i>	Durbin and Willshaw	Chen	Kirkpatrick <i>et al.</i>	Lin and Kernighan
(> 6.10)	5.90	5.70	5.80	5.70

Table 10.1.

These are typical results for a data set 50 towns spread at random over a square whose side is one unit long. The value given for the Hopfield algorithm has been obtained on a 30-town set and normalized to fit the 50-town problem. (We know that the length of the optimal tour that links N randomly distributed towns on a standard square map scales as \sqrt{N} .) The results of Kirkpatrick are given by a thermal annealing procedure applied to the classical 3-Opt optimization algorithm. It is thought that the Lin and Kernighan algorithm is the best classical procedure for the TSP problem to date. We can see that the Chen algorithm seems to be as good as that of Lin and Kernighan.

10.2.5 Ill-posed problems

Let us consider the set of equations

$$y_i = \sum_{j=1}^N a_{ij} x_j, \quad i = 1, 2, \dots, P,$$

or, in vector form,

$$\bar{y} = \mathbf{A} \cdot \bar{x}, \quad (10.39)$$

where \mathbf{A} is a full-rank $P \times N$ matrix, which means that

$$\text{rank}(\mathbf{A}) = \min(P, N).$$

The equation (10.39) is solved if it is possible to find a vector \tilde{x} such that the equality is satisfied.

- When $P = N$ the problem is well posed; a unique solution \tilde{x} exists.
- If $P < N$, that is to say if the number of equations is less than the number of unknowns, there exists a full subspace of \tilde{x} which satisfies the equations.
- If $P > N$, that is to say if the number of equations is larger than the number of unknowns, there is no solution.

In these last two cases the problem is an ill-posed one. But since there are no well-defined solutions one might want to determine the vector \tilde{x} which optimizes certain criteria.

Let us assume that $P > N$ and that one looks for solutions in real numbers by using analog neural networks.

The best solution is defined as the vector \tilde{x} which minimizes the quantity

$$H^{\text{cost}} = |\bar{y} - \mathbf{A}\tilde{x}|^2 = (\bar{y} - \mathbf{A}\tilde{x})^T \cdot (\bar{y} - \mathbf{A}\tilde{x}). \quad (10.40)$$

We have seen, in section 7.3.6, that the solution of this equation is given by

$$\tilde{x} = \mathbf{A}^+ \cdot \tilde{y},$$

where \mathbf{A}^+ is the pseudo-inverse matrix of \mathbf{A} , but here we devise a means of directly computing the vector \tilde{x} . This can be achieved by building a neural network with a dynamics driven by H^{cost} . Expanding the cost energy, one finds:

$$\begin{aligned} H^{\text{cost}} &= \tilde{y}^T \cdot \tilde{y} - 2\tilde{y}^T \cdot \mathbf{A} \cdot \tilde{x} + \tilde{x} \cdot \mathbf{A}^T \cdot \mathbf{A} \cdot \tilde{x} \\ &\simeq \sum_{ij} x_i (\mathbf{A}^T \cdot \mathbf{A})_{ij} x_j - 2 \sum_i (\tilde{y}^T \cdot \mathbf{A})_i x_i. \end{aligned}$$

The parameters of the network are given by

$$\begin{aligned} J_{ij} &= -2(\mathbf{A}^T \cdot \mathbf{A})_{ij}, \\ J_{i0} &= 2(\tilde{y}^T \cdot \mathbf{A})_i. \end{aligned} \tag{10.41}$$

As the space of solutions is convex one is certain that the network converges towards the unique solution.

With digital networks a difficulty arises, however, because the space of possible solutions is now restricted to the possible values of x_j . If the values of a_{ij} and y_i are also integers the equations to be solved are diophantic equations, a problem difficult to solve. However, one can consider that the natural ability for the neuronal networks to solve integer equations is a bonus rather than an inconvenience.

There is a way out. The binary expansion of an integer x_i is given by

$$x_i = \sum_{\alpha=1}^b 2^{\alpha-1} S_{i\alpha}, \quad S_{i\alpha} \in \{0, 1\}.$$

That is, a component x_i is represented by b neurons $\sigma_{i\alpha}$ instead of one. The cost energy becomes

$$H^{\text{cost}} = \sum_{i\alpha, j\beta} 2^{\alpha+\beta-2} S_{i\alpha} (\mathbf{A}^T \cdot \mathbf{A})_{ij} S_{j\beta} - \sum_{i\alpha} 2^\alpha (\tilde{y}^T \cdot \mathbf{A})_i S_{i\alpha}$$

and therefore the neuronal network is defined by the following parameters:

$$\begin{aligned} J_{i\alpha, j\beta} &= -2^{\alpha+\beta-1} (\mathbf{A}^T \cdot \mathbf{A})_{ij}, \\ J_{i\alpha, 0}^B &= 2^\alpha (\tilde{y}^T \cdot \mathbf{A})_i. \end{aligned}$$

According to their definitions, the x s are strictly positive numbers. One can extend the range of possible values by subtracting a constant value to the former definition of x_i .

- When $P < N$ there is an infinity of solutions. The problem is regularized by introducing further constraints which embed the assumed regularities of the solutions. The energy to be minimized becomes

$$H = |\tilde{y} - \mathbf{A}\tilde{x}|^2 + \lambda |H^{\text{cons}}(\tilde{x})|^2,$$

where H^{cons} is a constraint which must also be minimized. This technique has been applied to image processings. An image is a two-dimensional representation of a three-dimensional world. The problem of recognizing the nature of the items on the image is therefore essentially an ill-posed problem. However, the ambiguities of the

allocation can eventually be solved if the images are processed while using constraints associated with the regularities of the tri-dimensional space. This approach has been advocated by Poggio. The algorithms of Geman (see section 10.3.3) also proceed from the same philosophy. This technique has been applied, for example, to the treatment of seismic data (by Rothman) and proved to give reliable results for the detection of geological faults.

10.2.6 Final remarks on the application of neural networks to optimization problems

The optimization algorithms which are inspired by the theory of neural networks are more and more efficient. The algorithm of Chen for example gives better solutions to the TSP problem than the algorithms appealing to thermal annealing techniques while using CPU times that are shorter by a factor of five. The availability of dedicated neurocomputers would greatly increase this factor. However if it is certainly important to speed up the computational times by building neurocomputers, it is certainly even more important to realize that reasoning in the framework of the theory of neural networks may be a powerful source of new ideas.

It is also worth noting that the very meaning of optimization may not be the same for neural networks and for classical applied mathematics. Optimization problems are classified according to complexity classes. Easy problems are solved in a number of steps which scales as a polynomial of the size of the problem. For hard problems this number scales exponentially with the size. The class of NP problems is characterized by the fact that it is possible to decide in polynomial times if a solution does exist or not, even though no polynomial algorithm is known which yields the optimal solution. Living systems are, in general, not very demanding as regards the quality of the solutions they are looking for. Finding an approximate solution in short times is often of greater importance than striving to obtain a perfect response. For example, for a moving animal, finding a route essentially means avoiding obstacles while making progress towards the target, not looking for the path which would minimize the sugar consumption of its muscles. Therefore the distinction between easy problems and NP problems tends to be irrelevant for neural networks.

10.3 Low-level signal processing

10.3.1 Feature detectors

The first processing stages of sensory tracts aim at extracting features from sensory signals. For example lateral inhibition between retina cells enhances the contrast of visual signals between luminous and dark areas. The effect of these interactions somehow amounts to changing the image

into the Laplacian of the image. This may be carried out by two-layered feedforward networks made of input units ξ connected through short-range ‘Mexican hat’ interactions to output units σ . For example the transformation, on a square lattice, is given by

$$\sigma_{ij} = 4\xi_{i,j} - \xi_{i+1,j} - \xi_{i-1,j} - \xi_{i,j+1} - \xi_{i,j-1}. \quad (10.42)$$

This linear transformation gives a zero output for areas of input patterns which display a uniform activity (as for the background of an image for example). The following non-linear transformation is more interesting:

$$\sigma_{ij} = \text{sign}\left[4\xi_{i,j} - \xi_{i+1,j} - \xi_{i-1,j} - \xi_{i,j+1} - \xi_{i,j-1} - \theta_{ij}\right], \quad (10.43)$$

with $\xi \in \{-1, +1\}$. For $\theta_{ij} = 8 - \frac{1}{2}$ this equation yields $\sigma_{ij} = -1$, except when the input activities are $\xi_{i,j} = +1$ and $\xi_{i+1,j} = \xi_{i-1,j} = \xi_{i,j+1} = \xi_{i,j-1} = -1$. If the case arises, $\sigma_{ij} = +1$ and the system works as an ‘on-off’ feature detector. The detector is determined by the matrix

$$\mathbf{J}_{ij} = \begin{pmatrix} \cdot & -1 & \cdot \\ -1 & +4 & -1 \\ \cdot & -1 & \cdot \end{pmatrix} \quad (10.44)$$

of non-zero connections impinging on the neuron (ij) of the output layer. The matrix simply reproduces the pattern to be detected. The idea may be generalized to any sort of feature detector. For example, the following connection matrix with $\theta_{ij} = 16 - \frac{1}{2}$

$$\mathbf{J}_{ij} = \begin{pmatrix} -1 & -1 & +1 & +1 \\ -1 & -1 & +1 & +1 \\ -1 & -1 & +1 & +1 \\ -1 & -1 & +1 & +1 \end{pmatrix},$$

materializes an edge detector and

$$\mathbf{J}_{ij} = \begin{pmatrix} -1 & -1 & +2 \\ -1 & +2 & -1 \\ +2 & -1 & -1 \end{pmatrix},$$

with $\theta_{ij} = 12 - \frac{1}{2}$, a 45° line detector. This is exactly the sort of interaction that the synaptic dynamics of Linsker actually builds (see section 9.2.2).

The making of such ‘artificial retinas’ in silicon is under way.

It is interesting to note that straight line detection was introduced by people interested in character recognition, probably without being aware

that nature had invented a similar mechanism well before they had. The processing algorithm is called the Hough transform. The Hough transform maps a point (x, y) of one plane P^1 into a curve $\rho = \rho(\theta)$ in another plane P^2 . The curve is defined by

$$\rho = x \cos \theta + y \sin \theta.$$

The transformation of a line D of P^1 is a set of sine curves in P^2 which all cross at the same point determined by the parameters ρ^D and θ^D of the line (see Fig. 10.8). Let us assume that the signals in P^1 are thresholded: if the intensities of the representative points of a curve are below threshold and if the accumulated intensity at the crossing point is above threshold, the image of the line D in P^1 is a single point in P^2 . In this way the image of a set of lines is a set of points which can be called the signature of the set of lines.

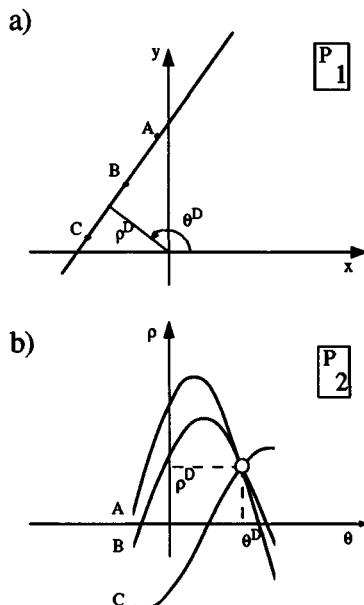


Figure 10.8. The Hough transform. The points A , B and C of a straight line D of plane P^1 give rise to sine curves which cross at the same point P of plane P^2 . If the signals of plane P^2 are thresholded the transformation maps D into a unique point P .

The Hough transform has been used, for example, for the recognition of handwritten Hebraic block capitals. Square Hebrew is mainly made of

straight lines and a letter gives rise to a specific signature. Recognition which uses a classical pattern-matching algorithm performs well. It would be interesting to devise a neural network which would carry out both the feature extraction and the pattern matching.

This example of a convergence between an algorithm imagined by computer scientists and the natural mechanism one observes in the visual system is an encouragement to look for other types of features which the natural networks extract from the visual signals and to study how they are processed in the cortex.

10.3.2 A network for automatic image matching

The eyeballs continuously move in the orbits even though the gaze is fixed on a precise point of the environment.

The wobbling amplitude is of the order of one degree, whereas it is possible to separate two points the angular distance of which is of the order of one minute of arc. This problem was tackled by Van Essen who proposed a solution appealing to a neural network similar to the circuits which are used in telephone networks.

A comprehensive telephone network must be able to connect each pair of speakers. The most straightforward network is the crossbar circuit, which needs N^2 relays. The Ω -network achieves the same goal by using only $N \times \log N$ relays provided that the lines can be multiplexed. The Ω -network is made of $\log(N) + 1$ layers. There would exist such a system of layers and synapses (relays) in the nervous system which would shift the image of one eye layer by layer leftwards or rightwards until the coincidence between the signals coming from both eyes is maximal (see Fig. 10.9). Van Essen argues that the system of layers could be found either in the layered structure of the lateral geniculate nuclei (LGN) or in the sublayers (IVa, ..., IVc) of the fourth lamina of the visual area VI.

10.3.3 Image restoration

Let I be an image which has been degraded owing to transmission errors or to defects which are inherent to the technical process, such as photography, used in the making of the image. If a bit of a line of the image is missing somewhere it is likely that the missing bit is a defect, not a distinctive feature of the image. This last sentence is a statement concerning what images should look like. It embeds some preknowledge of the universe of images one has to deal with. Preknowledge makes some arrangements of features seen in the image very unlikely. They can be modified without running the risk of introducing spurious information into the image. Likely arrangements, on the other hand, have not to be touched. Modifying the image piece by piece according to the likely

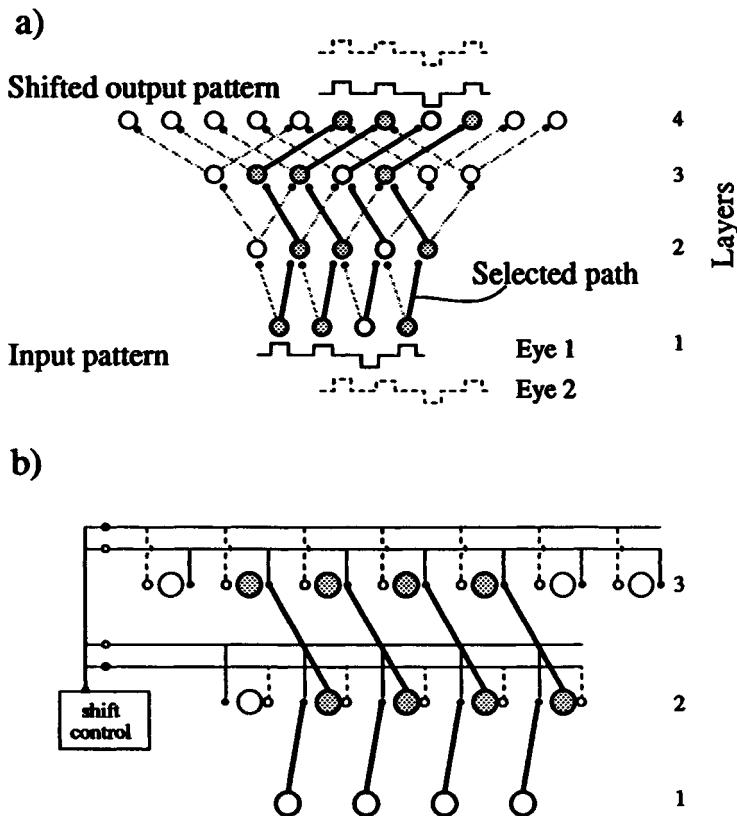


Figure 10.9. A bifurcation system which automatically aligns the images coming from both retinas. The lower part of the figure shows a possible implementation of the routing mechanism which uses inhibitory synapses (After Van Essen).

arrangement determined by preknowledge leads to a restored image. In the Geman algorithm an energy is associated with various feature arrangements and the likelihood of the arrangements is given by the Gibbs distribution determined by the energies. Let us give two examples:

- We receive a degraded geographical political map of some part of the old world and we want to have it restored. Because this is a political map we know that it is made of regions of uniform intensities, say of uniform grey levels (Fig. 10.10.a). Therefore if the intensity of a pixel i is

$$\sigma_i = k, \quad k = 1, \dots, K \text{ grey levels},$$

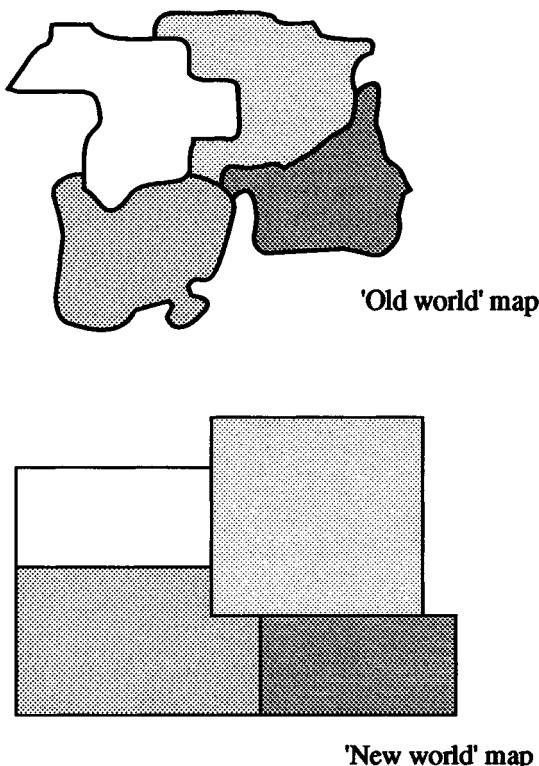


Figure 10.10. Two political maps, one of the Old World countries and one of the New World countries. The knowledge conveyed by the term ‘political map’ is that a country is given a uniform color. The knowledge associated with ‘New World’ is that borders are likely to be straight lines.

it is likely that the intensities of the surrounding pixels are the same. Let us call \mathcal{V} the neighborhood of a pixel i . For example \mathcal{V} is the set of the eight neighbors j of i . Then the (two-bodies) energies H_{ij}^0 between the pixel states are given by

$$H_{ij}^0 = \begin{cases} 0 & \text{if } j \notin \mathcal{V}, \\ 1 & \text{if } j \in \mathcal{V} \text{ and } \sigma_j \neq \sigma_i, \\ -1 & \text{if } j \in \mathcal{V} \text{ and } \sigma_j = \sigma_i. \end{cases} \quad (10.45)$$

This defines an energy function for the image. The modifications are

carried out by using the Gibbs distribution,

$$\rho(I = \{\sigma_i\}) = \frac{1}{Z} \exp\left(-\sum_{ij} \frac{H_{ij}^0}{B}\right),$$

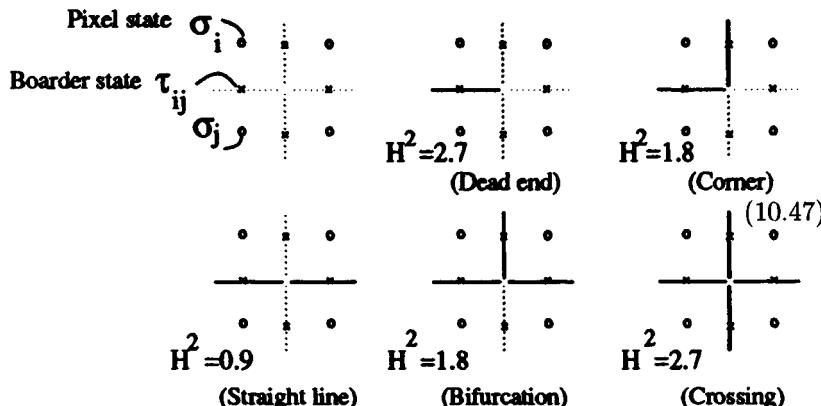
where the ‘temperature’ B is slowly decreased as is usual in the thermal annealing procedures.

- In the second example we receive a geographical political map of some region of the New World. The information conveyed by the words ‘New World’ is that the borders are likely to be straight lines. There are now two sorts of pixels associated with two lattices. The first lattice is the square lattice of pixel intensities σ_i which we introduced above. The interactions between these pixels are given by Eqs. (10.45). The second lattice is that of the links $\tau_{(ij)}$ corresponding to edges (ij) of the first lattice (see Fig. 10.10.b). If a segment of a border is present at some site (ij) of this lattice the corresponding state is $\tau_{(ij)} = 1$. It is $\tau_{(ij)} = 0$ otherwise.

Geman and Geman associate the following (one-body) energies $H_{(ij)}^1$ to the various situations which can be encountered:

$$H_{(ij)}^1 = \begin{cases} 0 & \text{if } \tau_{(ij)} = 1 \text{ whatever } \sigma_i \text{ and } \sigma_j, \\ H_{ij}^0 & \text{if } \tau_{(ij)} = 0, \end{cases} \quad (10.46)$$

because a border breaks the necessary continuity between the grey levels. They also introduce the (many-bodies) energies H^2 which are depicted in the diagram (10.47) to account for the tendency of borders to make straight lines.



Those algorithms work fairly well. It remains to be seen how they could be implemented into neural networks. A major problem involves

that of understanding how preknowledge is memorized and how it is used by the network to correct the assumed defects of the sensory signal. It must be emphasized that the problem here is not one of pattern matching but of the transformation of an image which possibly has never been seen before into an image which fits the previous knowledge regarding for example the regularities of the outside world. This corrected pattern can be memorized as a new pattern.

10.4 Pattern matching

10.4.1 Parallel processing of images

The visual tract extracts several features in parallel. These features make patterns which are also processed in parallel and then merged to give the visual scene its meaning (in actual fact this last point is still to be experimentally proved). Experimental observation shows however that some serial process also takes place when the network has to recognize whether two patterns are simultaneously present in a picture. It must be stressed that parallel and serial treatments of patterns are not contradictory to each other because they involve mechanisms which are different in nature. The mechanism which drives the serial search is an active process so to speak (the system has to check successive hypotheses one by one) whereas the one which is at stake in a parallel process is passive.

Cerný proposed a network which is able to combine poor information coming from various channels so as to reach a reliable conclusion regarding the meaning of the incoming signals.

The system is made of a number K of channels, each receiving a badly damaged version I of an unknown input pattern. There are P possible input patterns I^μ and the system has to identify the pattern which is the source of the inputs. The K automata are arranged in a one-dimensional array. The possible states σ_k of an automaton k are given by (Fig. 10.11)

$$\sigma_k = \mu, \quad k = 1, 2, \dots, K \quad \text{and} \quad \mu \in \{1, 2, \dots, P\}.$$

The ‘energy’ of the automaton k in state σ_k is made up of two terms:

- a local term,

$$H^0(\sigma_k) = I \cdot I^{\mu(=\sigma_k)},$$

which is related to the Hamming distance between input I and the prototype I^μ corresponding to the internal state σ_k of automaton k ; and

- a first-neighbor interaction term,

$$H^1(\sigma_k) = 0, 1 \text{ or } 2,$$

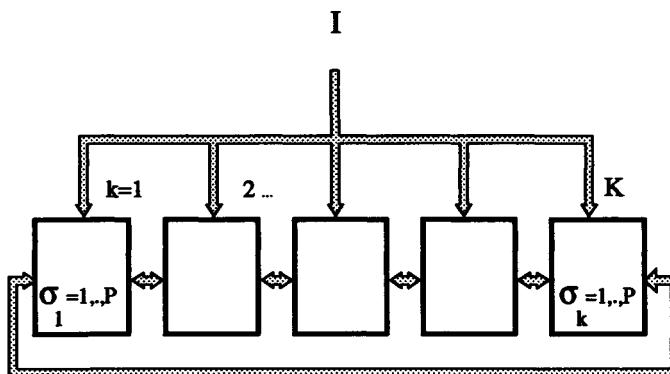


Figure 10.11. The array processor of Cerný.

which is the number of neighbor automata the states of which are not the same as that of the state of k . The dynamics is that of thermal annealing using a Metropolis algorithm. The state of an automaton is changed at random and the variation ΔH of the energy is computed.

- If $\Delta H < 0$ the change is accepted.
- If $\Delta H > 0$ it is accepted with a probability given by,

$$\bar{\omega} = \exp\left(-\frac{\Delta H}{B}\right).$$

The system finds the solution in a short time even though the inputs are severely damaged.

This is an interesting model and much work can be done on it. For example, it would be worth trying to have it implemented in a neural network (this seems to be easy). It would also be interesting to determine the memory storage capacity of this system. Finally, it must be stressed that the architecture of the model of Cerný, although it is very simple, is relatively close to the modular structure of the cortex.

10.4.2 Pattern matching in fully connected neural networks

The most straightforward application of fully connected neural networks is pattern matching. One knows a number of patterns I^μ which are called prototypes. The signal I^0 is a blurred or distorted version of one of the prototypes. The problem is to identify the prototype which the signal is derived from. This is a direct application of the associative theory of neural networks. The patterns are stored in the synaptic efficacies of a fully connected network.

The signal I^0 is the ‘percept’, the initial condition of the neural dynamics $I^0 = I(t = 0)$, and the prototype, the fixed point where one arrives at $I^* = I(t = \infty)$, is the ‘concept’.

A classical program, one which is processed by ordinary computers, involves the calculation of P scalar products $I^* \cdot I^\mu$ between the signal and the P various prototypes I^μ . If n_a and n_m are the number of ticks necessary for the computation of addition and of multiplication respectively, the time required by the calculation is

$$(n_a + n_m) NP.$$

It is then necessary to compare these P scalar products with one another to pick up the largest one. The time needed for this last operation scales as P . Fully connected neural networks converge in about one refractory period and therefore they can outdo the serial machines by several orders of magnitude. Large, fully connected networks are difficult to make, however. In Chapter 11 semi-parallel architectures are described. They are much more feasible than the fully connected networks. The convergence times of semi-parallel machines scales as N , the number of neurons, and therefore they outdo the serial machines by a time factor which scales as P the number of patterns.

Various applications of pattern matching have been proposed. For example, a machine using optic fibers to identify planes from radar images has been assembled. Neural networks have also been used to recognize separate handwritten characters (Dreyfus, Jackel).

Image compression is a possible interesting application where one can take full advantage of the inherent parallelism of neural networks together with the ability to carry out swift pattern matching (Denker). Image compression consists of decomposing the image I in a set of partial images I^k , $k = 1, 2, \dots, K$. Every sub-image I^k is compared with every pattern of a given set of standard patterns of identical sizes I^μ , $\mu = 1, 2, \dots, P$. Let $\mu(k)$ be the label of the standard pattern I^μ which the sub-image k most resembles. Instead of sending every bit of I , it is much more economical to send the string $(\mu(1), \mu(2), \dots, \mu(K))$ of labels through the transmission channel. On receiving the signal, the image is reconstructed by using the same set of standard patterns. It is possible to improve the efficiency of the process by appealing to hierarchical structures of standard patterns. Image compression can also be carried out by using expert neural networks (see section 10.6.2 for a brief account of expert neural networks).

In many applications, of which image compression is an example, one only needs the label of the pattern closest to the input vector. When the number P of stored patterns is significantly smaller than the number N of neurons it is worthwhile to appeal to grandmother cells networks. Such a network has been proposed by L. Jackel, H. Graf and R. Howard. It is made of a P neuron neural network N^0 fed by an N neuron feedforward

network N^I . Such a network is also called a Hamming network (see Fig. 10.12). The $N \times P$ connection matrix of network N^I is a template matrix of memorized patterns. Its elements are given by

$$J_{i\mu} = \xi_i^\mu, \quad i = 1, 2, \dots, N, \quad \mu = 1, 2, \dots, P.$$

The elements of the $P \times P$ connection matrix of N^0 are all inhibitory:

$$J_{\mu\nu} = -1, \quad \mu = \nu = 1, 2, \dots, N.$$

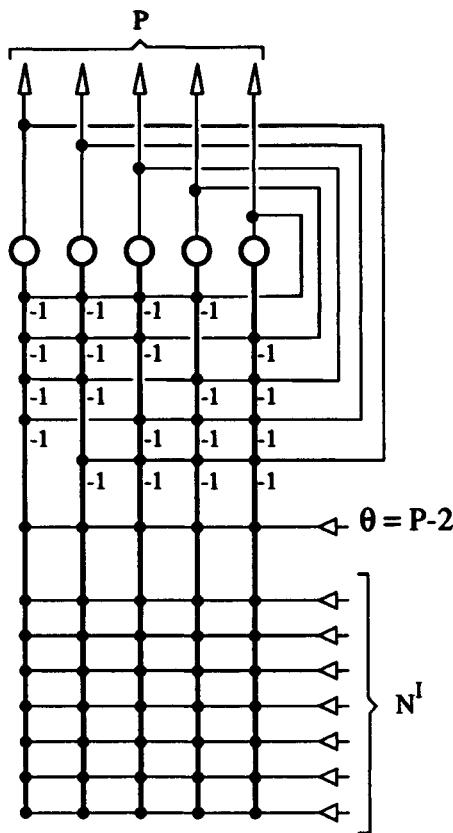


Figure 10.12. A Hamming net. It is made of two pieces, an input network for matching and an output network which wipes out all activities but the largest one.

The thresholds of neurons are set to a value which ensures that only one neuron, the grandmother neuron of pattern μ , can fire at a time.

N^0 is therefore a ‘winner takes all’ circuit. The dynamics proceeds as follows:

- The input vector gives rise to average neuronal activities which depend on the overlap between the incoming and the memorized patterns. The larger the overlap the larger the activity.
- The network N^0 kills all activities but the largest one.

It must be stressed that this network demands non-zero noise levels or the use of analog devices (see Chapter 11). It also stores little information.

10.4.3 Pattern invariances and graph recognition

Invariant recognition is one of the most challenging problems of AI. How does the brain manage to recognize so easily even severely distorted patterns, whereas recognition algorithms must appeal to somehow complex mathematical transformations to cope with such simple modifications as translations, rotations or dilatations of the objects to be classified?

This problem can be tackled in its full generality: a pattern is a graph, determined by a set of vertices and a set of bonds linking some of the vertices. A permutation of the vertices (a homeomorphism) generates a new graph. Homeomorphisms comprise all sorts of transformations, translations, rotations, dilatations but also affine transformations, foldings, . . . All these transformed patterns are considered as belonging to the same (equivalence) class, that defined by the source pattern. If several source patterns are memorized and if the network experiences a graph, the problem involves that of retrieving the source graph it comes from. Here we present the solution of Kree and Zippelius.

Let $i, i = 1, 2, \dots, N$ label the vertices of a network. A pattern is a graph G which is determined by a set of bonds (Fig. 10.13). This set is represented by an $N \times N$ matrix \mathbf{G} , the adjacency matrix of the graph G :

$$\mathbf{G}_{ij} = \begin{cases} 1 & \text{if there exists a link between } i \text{ and } j, \\ 0 & \text{otherwise.} \end{cases}$$

• We consider that the matrix elements \mathbf{G}_{ij} describe some input ‘retinal’ neural activities. This is an interesting approach since we have seen that there exists a cortical process which decomposes retinal inputs into a set of line orientations (which may be identified with the edges of the graph). On the other hand the distance between two patterns (two graphs) G^1 and G^2 is the number of bonds which are not shared by the two patterns. This is the Hamming distance between the two representative strings of activities in the retina. This distance is given by

$$[d(G^1, G^2)]^2 = \frac{1}{2} \text{Tr}[(\mathbf{G}^1 - \mathbf{G}^2)^2]. \quad (10.48)$$

- A permutation π of the N vertices is represented by an $N \times N$ matrix $\mathbf{T}(\pi)$ with elements

$$\mathbf{T}_{i\alpha} = \begin{cases} 1 & \text{if } \alpha = \pi(i), \\ 0 & \text{otherwise.} \end{cases}$$

This is a matrix with columns and rows comprising one and only one non-zero element each. We introduce a neural network called a preprocessor. It comes in the form of an $N \times N$ lattice of neurons labeled (i, α) whose activities $\mathbf{T}_{i\alpha} \in \{0, 1\}$ represent a

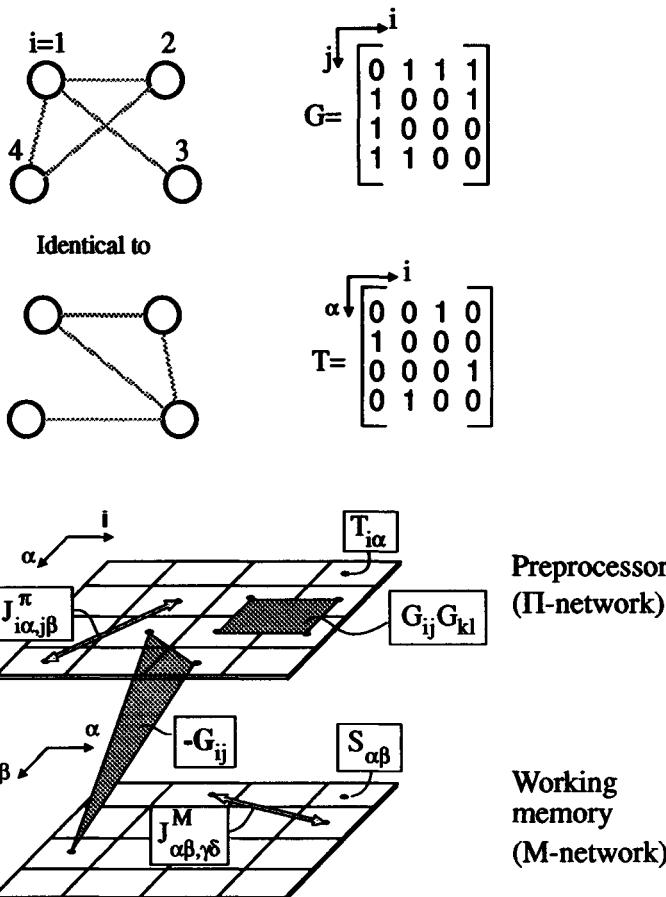


Figure 10.13. The architecture of the network put forward by Kree and Zippelius to account for invariant pattern recognition.

permutation π of the vertices i of G . According to the results of section 10.2.2, this can be achieved by a neuronal dynamics which is driven by the energy function

$$H^\pi(\mathbf{T}) = \sum_i \left(1 - \sum_\alpha T_{i\alpha}\right)^2 + \sum_\alpha \left(1 - \sum_i T_{i\alpha}\right)^2, \quad T_{i\alpha} = 0, 1,$$

which is to be identified with

$$H^\pi(\mathbf{T}) = - \sum_{(i\alpha, j\beta)} J_{i\alpha, j\beta}^\pi T_{i\alpha} T_{j\beta} - \sum_{i\alpha} J_{i\alpha, 0}^\pi T_{i\alpha},$$

yielding

$$J_{i\alpha, j\beta}^\pi = -2(\delta_{\alpha\beta}(1 - \delta_{ij}) + \delta_{ij}(1 - \delta_{\alpha\beta})) \quad \text{and} \quad J_{i\alpha, 0}^\pi = +2. \quad (10.49)$$

- There is a second neural network, comprising $N \times N$ neurons, which stores the patterns, the prototype graphs S^μ , to be memorized in the form of adjacency matrices S^μ . The graphs are generally simple with a small number of bonds: the patterns are therefore strongly correlated and one has to appeal to one of the symmetrized, perceptron-like learning rules to have all the prototype patterns correctly stored. The energy function associated with that network in state S is

$$H^M(S) = - \sum_{\alpha\beta} \sum_{\gamma\delta} J_{\alpha\beta, \gamma\delta}^M S_{\alpha\beta} S_{\gamma\delta}, \quad S_{\alpha\beta} \in \{0, 1\}, \quad (10.50)$$

where the matrix elements $J_{\alpha\beta, \gamma\delta}^M$ (including the thresholds $J_{\alpha\beta, 0}^M$) are given by the learning dynamics.

- Finally, the π -network with dynamic variables $T_{i\alpha}$ is related to the M -network with dynamic variables $S_{\alpha\beta}$. The goal of the system is, given an input graph G (that is, a fixed set of activities in the retina), to find a permutation π and a pattern I^μ such as the distance between the transformed graph,

$$T(G) = T^T \cdot G \cdot T,$$

and the prototype graph $I^\mu = \{S_{\alpha\beta}^\mu\}$ is zero.

The distance between $T(G)$ and S is given by

$$H^C(S, T) = d(T(G), S)^2 = \frac{1}{2} \text{Tr}[(T^T \cdot G \cdot T - S)^2], \quad (10.51)$$

which may be written as

$$H^C = \frac{1}{2} \sum_{\alpha, \beta} \left(\sum_{i, j} T_{i\alpha} T_{j\beta} G_{ij} - S_{\alpha\beta} \right)^2.$$

H^C can be thought as a cost function, an energy which has to be minimized. The total energy is therefore

$$H = \lambda H^\pi(T) + \mu H^M(S) + H^C(S, T), \quad (10.52)$$

where λ and μ are adjustable Lagrange parameters. When the energy is zero the system has found a solution, which means that it has both found a permutation which transforms the input graph into one of the memorized patterns and that it has found that memorized pattern. The last term of Eq. (10.52) couples the preprocessor and the memory lattices. It introduces terms of 4th order ($(G_{ij})^2 = G_{ij}$),

$$G_{ij} G_{k\ell} T_{i\alpha} T_{j\beta} T_{k\alpha} T_{\ell\beta},$$

in the preprocessor network and terms of third order,

$$-G_{ij} T_{i\alpha} T_{j\beta} S_{\alpha\beta},$$

between the preprocessor and the memory networks.

The solution, as usual, can be looked for by using the thermal annealing technique. However, there is a problem: the energy of the system scales as N (which is convenient) whereas the entropy scales as

$$S = \log(N!) = N \log N$$

instead of N , and the system remains in the high-temperature phase, which is not what is desired. To solve the problem the temperature must scale as $1/\log N$. This is a problem we have already encountered in combinatorial optimization.

This model, at the time of writing this section, had not been verified but it is no doubt of too general a nature. Badly distorted graphs cannot be recognizable even by a trained eye and they can nevertheless be classified by the system. This could be improved by restricting the permutations to some types of legitimate moves.

10.4.4 Processing auditory signals: the time warping problem

The auditory signals are preprocessed through the auditory tract. In particular the cochlea system of the inner ear transforms the sound waves in amplitude frequency coded patterns. These patterns can be treated in very much the same way as the visual patterns. There are specific problems however which are listed below.

- Patterns are essentially phonemes. How can a particular string of phonemes be recognized as a certain word?
- Speaking is a continuous flow of sound signals with no specific marker between the phonemes and no marker between the words. How can words be recognized as separate entities?
- The speed used by a given speaker to utter his sentences changes continuously along his speech. It also varies from one speaker to the other. This is the problem of time warping. How can a neural network manage to ‘unwarp’ the auditory signals?

These difficulties are solved by a neural network proposed by Tank and Hopfield. It is made up of two parts (see Fig. 10.14):

1) A fully connected network of neurons i with fixed inhibitory interactions $J_{ij} = -1, J_{ii} = 0$. The number of neurons is the number of words N_w which the system can recognize. The neurons i are grandmother cells: all neurons are silent except one that fires when the system experiences a particular word.

2) An input system. Every input is associated with a particular phoneme. It makes contact with the neurons of the main circuit through sets of increasing delays.

Let $\nu_i^x = 1, 2, \dots$ be the delay for the phoneme X on the input line of i . A word W is a string of phonemes X . Let for example two words $W1$ and $W2$ be defined by (from left to right)

$$W1 = X3 X4 X1 X2, \quad W2 = X2 X2 X3 X1.$$

The dynamics of the system is determined by the local fields

$$h_i(\nu) = - \sum_j S_j(\nu) + h_i^{\text{ex}}(\nu) - \frac{1}{2} N_w,$$

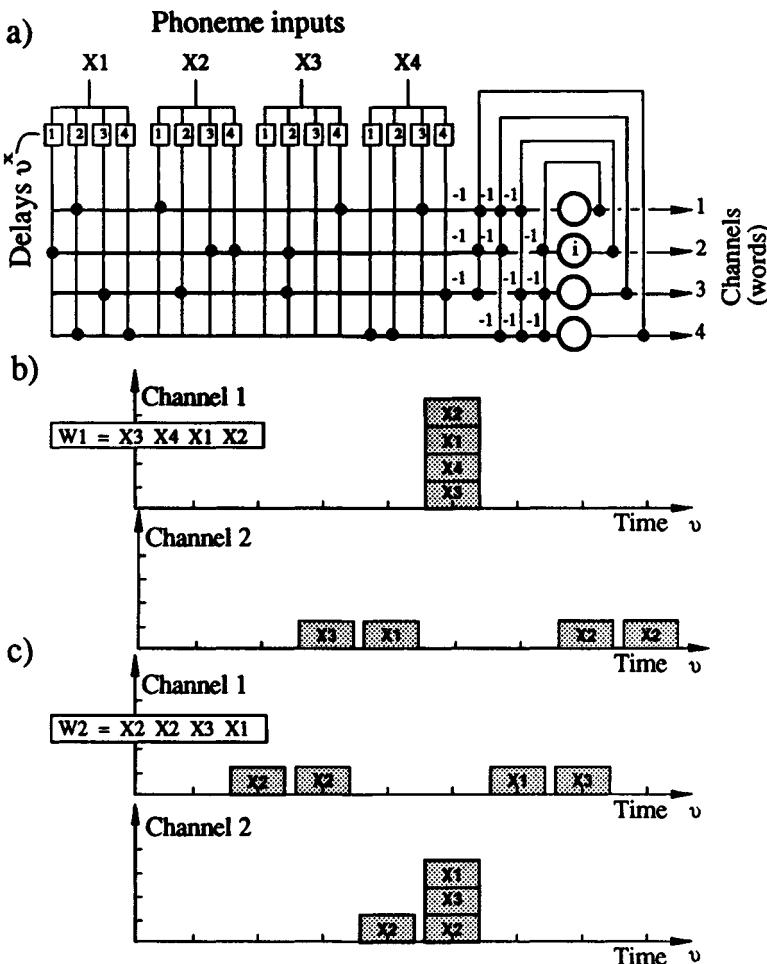


Figure 10.14. a) The network of Tank and Hopfield for speech recognition is made of delay lines feeding a winner takes all circuit. The circuit selects the right word. X_i are phonemes.
 b) and c) The figures show how the signals pile up in the delay lines to sort out the memorized words.

with

$$h_i^{\text{ex}}(\nu) = \sum_x \xi_i^x (\nu - \nu_i^x)$$

and

$$S_i(\nu + 1) = \mathbf{1}(h_i(\nu)), \quad S_i \in \{0, 1\}.$$

The delays ν_i^x for one input line i , that is for one word W_i , are set so

as to make the signals elicited by the various phonemes of the word W_i arrive at the same time on neuron i : the input field h_i^{ex} is then as large as possible. The signals on the other lines arrive at random and the corresponding neurons are silent.

Let us now assume that the signal is made of two words W_1 and W_2 :

$$W = W_1 W_2.$$

The system starts recognizing the first word W_1 and the neuron $i = 1$ fires while all the other neurons remain silent. Then the system experiences a mixture of the two words, the end of the first word and the head of the second word. The signal is

$$W' = X_1 X_2 X_2 X_2.$$

The excitations are now unable to trigger the firing of any neuron. Afterwards the elementary excitations due to the word W_2 add up in the line of neuron $i = 2$, which starts firing. The system therefore succeeds in separating the two words W_1 and W_2 .

The problem of time warping is solved by using memory functions instead of well-peaked delays. The external field is given by

$$h_i^{\text{ex}}(t) = \sum_x \int dt f_k(\tau) \xi_x(t - \tau), \quad k = \nu_i^x,$$

with, for example,

$$f_k(\tau) = e^{-n} \left(\frac{\tau}{k} \right)^n \exp \left(-n \frac{\tau}{k} \right) \quad (n \text{ given}).$$

With such a preprocessing the system is still able to separate words which are distorted by time warping. An analytical study of the recognition process can be carried out by introducing a time-dependent energy given by

$$H^c(t) = \sum_{\langle ij \rangle} S_i S_j + \sum_i \left(\frac{1}{2} N_w + h_i^{\text{ex}}(t) \right) S_i.$$

10.5 Some speculations on biological systems

10.5.1 The function of the cerebellum

One of the earliest models of neural network, due to Kohonen, is a two-layer, feedforward, linear system. The system associates an N^I -unit input pattern $I^{\mu, \text{in}}$, $\mu = 1, 2, \dots, P$ with an N^O -unit output pattern $I^{\mu, \text{out}}$.

The neuronal states are analog signals σ_i . It is assumed that the vectors $\tilde{\xi}^{\mu, \text{in}} = \{\xi_i^{\mu, \text{in}}\}$ and $\tilde{\xi}^{\mu, \text{out}} = \{\xi_i^{\mu, \text{out}}\}$ are normalized:

$$|\tilde{\xi}^{\mu}|^2 = \sum_i (\xi_i^{\mu})^2 = 1.$$

The synaptic efficacies are Hebbian and given by

$$J_{ij} = \frac{1}{N^{\text{in}}} \sum_{\mu} \xi_i^{\mu, \text{out}} \xi_j^{\mu, \text{in}}.$$

The (linear) response of the system to a given input I^{ν} is

$$\begin{aligned} \sigma_i(I^{\nu}) &= \sum_j J_{ij} \xi_j^{\nu, \text{in}} = \frac{1}{N^{\text{in}}} \left[\sum_j (\xi_j^{\nu, \text{in}})^2 \xi_i^{\nu, \text{out}} + \sum_{\mu \neq \nu} \xi_i^{\mu, \text{out}} \xi_j^{\mu, \text{in}} \xi_j^{\nu, \text{in}} \right] \\ &= \xi_i^{\nu, \text{out}} + O((PN^{\text{in}})^{-1/2}), \end{aligned}$$

which is a slightly blurred version of the state $I^{\nu, \text{out}}$ associated with $I^{\nu, \text{in}}$ provided that the patterns are not too correlated. (To deal with correlated patterns Kohonen introduced Penrose pseudo-inverse matrices as explained in section 7.3.6.)

This direct association between a complex input pattern of activities and a complex output pattern of responses has been taken as a possible, albeit extremely simplified, model of cerebellum, in particular by Pellioniz and Llinas. If the input and output cells are topologically ordered the system provides a mapping between a geometrical space of inputs and a geometrical space of outputs. The structure of the cerebellum was described in section 2.2.7. We have seen that indeed the cerebellum is a very well organized system comprising a few types of neuronal cells.

In actual fact the mechanism proposed by Marr is slightly different: instead of taking place through classical homosynaptic (axo-dendritic) junctions, the association process is achieved through hetero-synaptic contacts. There are two sets of afferents on the dendritic trees of Purkinje cells, the parallel fibers and the climbing fibers. Association is between the patterns conveyed by these two types of afferents. Temporal coincidences between the activities of both types of fibers would lower the output of the Purkinje cells, while bad temporal fittings would enhance the signals. The output of Purkinje cells would then be an error signal compensating for the discrepancy between a model motion and the actual motion.

10.5.2 The function of the hippocampus

The hippocampus is a structure which plays an important role in the long-term memorization process. Its structure is also very regular and

a striking feature is a set of cells, the pyramidal cells, which receive information from the cortex through two different pathways (see section 2.2.7). The pathways interfere on the dendrites of the pyramidal cells, making hetero-synaptic junctions in very much the same way as fibers interfere on the Purkinje cells of the cerebellum. It has been shown that these junctions are modifiable during learning.

The polarizing field on the pyramidal cell k can be written as

$$h_k = \sum_{ij} J_{ijk} \sigma_i \sigma_j,$$

where i and j label two cortical cells impinging on k . One again assumes that the learning rule of these contacts is Hebbian:

$$\Delta J_{ijk} \simeq \varepsilon \xi_i^\mu \xi_j^\mu, \quad \varepsilon > 0.$$

One notes that ΔJ_{ijk} does not depend on k . The polarizing field of k gives a measure of how well a pattern I^μ is already imprinted in the system: according to this scheme the hippocampus carries out self-comparisons of cortical activities. The pyramidal cell activities in turn can control the memorization processes of the cortex through the emission of neurochemicals or by the modification of the threshold of cortical cells (Fig. 10.15).

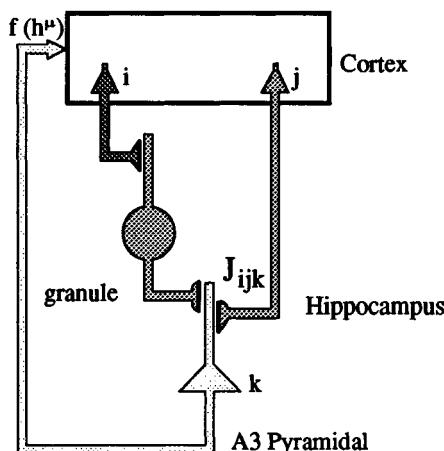


Figure 10.15. A putative mechanism of the role of hippocampus in the process of long-term memorization.

Therefore the modifications of cortical synaptic efficacies are driven by a Hebbian learning rule which is weighted by a factor f :

$$\Delta J_{ij}(\mu) = \varepsilon \xi_i^\mu \xi_j^\mu f(h(\mu)), \quad \varepsilon > 0, \quad (10.53)$$

where f is given for instance by

$$f(h^\mu) \simeq 1 - S(h_k(\mu)).$$

$S(x)$ is the response function of neuron k and

$$h_k(\mu) = \sum_{ij} J_{ijk} \xi_i^\mu \xi_j^\mu.$$

If the hippocampus is destroyed the long-term memorization process can no longer take place as it is observed. It is worth noting that the form of the learning rule (10.53) is exactly the one we put forward for the storing of patterns with optimal efficiency (see section 7.4.3).

It is likely that the hippocampus is involved in other functions as well. The process of memorization would imply the granule cells and the pyramidal cells A3. But the architecture of the hippocampus also comprises A1 pyramidal cells. A suggestion of Rolls is that the role of these structures would be to allocate the patterns to be imprinted, among the various regions of the cortex. Time effects would also be involved in the hippocampus owing to the regularly increasing size of its cells in the dentate gyrus for example. Finally, hippocampus is the locus of a periodic activity called the theta rhythm whose function is not clear. The fact that rats tend to synchronize their olfactory intake with the theta rhythm of their hippocampus could yield an indication.

10.5.3 Attention: the searchlight hypothesis

We have seen in section 2.4.6 that attention is controlled by the reticular formation. Interesting experiments have been carried out by Treisman on the one hand and Julesz on the other which have shown that the recognition of conjunctions of patterns is a sequential process. For example, one recognizes in less than a hundred milliseconds a letter among a set of other letters. However, if the letters are colored and if one is asked to find a letter of a given color, that is if one is asked to recognize a combination of color and shape, the search takes a much longer time. Moreover, the larger the number of items, the longer the search. It thus appears as if the activity is first focused on shape recognition and afterwards on color association.

Crick has put forward the following mechanism for attention: the activity in the cortex is not uniform. A region of slightly higher than

average activity, say R , would trigger the excitation of the afferents of the reticular formation which are geometrically associated with that region. Due to the fact that interactions between the neurons of the reticular formation are inhibitory, all the neurons of the formation, except those associated with the region R , become silent. Let us assume that the effect of the reticular activity is to control the threshold level of cortical neurons. Then all regions of the cortex except R , which is the attentive region where the signals can still be processed, are silent. The activity of the cortex is limited to R . As a consequence the overall average activity of the cortical tissue is considerably reduced, a fact that is supported by experimental observation.

This situation cannot be stable however and the searchlight must move to other areas to enable the cortex to process other information. Therefore, according to Crick, there is a mechanism which can be found in the thalamus, which forces the region R to move towards other locations in the cortex.

The explanation of attention phenomena by Crick is only qualitative. It would be interesting to devise networks which could display this dynamics.

10.5.4 The role of dream sleep

We have emphasized that a simple Hebbian learning dynamics leads to an overcrowding catastrophe. It has also the inconvenience of storing the patterns unevenly since a pattern which is experienced twice as long as another pattern is imprinted twice as much. Therefore a strongly imprinted pattern due to a striking experience during the waking state of an individual might become obsessional: its basin of attraction could be so large that it could trap all possible percepts. According to Crick and Mitchinson, the role of dream sleep would be to make the basins of attraction of the memorized patterns equally large.

Hopfield suggests that dream sleep resets the state of the system at random from time to time. The neural state then evolves according to the usual dynamics and reaches a fixed point. If the system is obsessed by a certain pattern the most probable fixed point I^μ is that pattern. Then the synaptic efficacies are modified according to a Hebbian anti-learning rule:

$$\Delta J_{ij} = -\varepsilon \xi_i^\mu \xi_j^\mu, \quad \varepsilon > 0.$$

This makes the basins of the most frequently visited patterns shallower and therefore the antilearning process tends to make the set of basins more even. To support these views Crick and Mitchinson argue that not all animal species are dream-sleep prone. The animals which do not dream, as revealed by EEG's, tend to have cortices out of proportion with

what they are able to do; an example is the Australian spiny anteater. This would be due to the necessity for this animal to store a number of patterns without being able to optimize the storing process.

This explanation, clever though it may seem, is not really satisfactory. For example the antilearning procedure must not be pursued too far because it would eventually erase all basins of attraction and all memory would be lost. On the other hand we have seen alternative explanations such as the optimal storing procedures which automatically ensure that all basins are of about the same size (in section 4.5.3).

10.6 Higher associative functions

10.6.1 Application of topological maps to robotics

The Kohonen learning algorithm which we introduced in section 9.1.3 builds a mapping between the (continuous) space $\tilde{\xi}$ of input activities and the D -dimensional output neural space: after training a neuron, called the ‘most sensitive’ neuron, is associated with every vector $\tilde{\xi}$. This vector is given the label of the neuron under scrutiny.

It happens that the neighborhood relations in the D -dimensional neural network are preserved in the space of inputs $\tilde{\xi}$.

Kohonen considers a system made of two arms and one eye in front of a two-dimensional working surface (see Fig. 10.16). The tips of the two arms and the gaze of the eye all meet on a given point P of the working surface. The space of inputs $\tilde{\xi}$ is six-dimensional. The parameters are the position of each arm ξ^1, ξ^2 and ξ^3, ξ^4 and the direction ξ^5, ξ^6 of the gaze. The neural network is two-dimensional ($D = 2$) and the neurons are at the vertices of the square lattice. The training session consists in associating the six-dimensional vectors $\tilde{\xi}$ with the two-dimensional vector that determines P by using the topological algorithm of section 9.1.3. After training one finds a topological map for every two pairs of input variables. Let us choose a point P of the working surface and let us feed the input lines of the neural network with the two parameters ξ^5 and ξ^6 , those associated with the eye. These parameters determine a neuron, the most sensitive neuron $i_c(\xi^5, \xi^6)$. In reality i_c is simply the neuron which lies in the gazing direction. The important point is that the very same neuron i_c is most sensitive for the parameters ξ^1 and ξ^2 of the first arm and for parameters ξ^3 and ξ^4 of the second arm which corresponds to the same point P of the working surface. Therefore a neural network which modifies the parameters of the eye so as to make a certain neuron at position P as sensitive as possible, automatically positions the two arms on point P , that is on the point corresponding to the gaze. The Kohonen approach is now being applied to maps with constraint, those

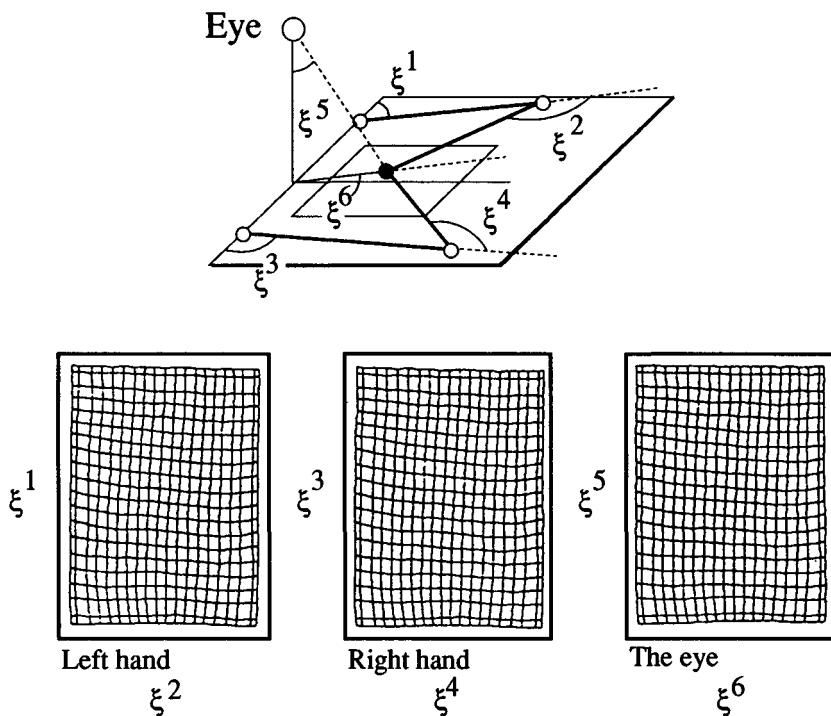


Figure 10.16. The two-arms robot automatically places the tips of the arms at its gazing point (After Kohonen).

associated with the occurrence of obstacles on the trajectories of moving parts of a robot, for example.

10.6.2 Neural networks and expert systems

An expert system is made of a data base, which is a set of logical propositions such as

$$\text{if } A \text{ then } B \quad (\text{Inf. 1})$$

or

$$\text{if } A \text{ then } B \text{ or } C \quad (\text{Inf. 2}),$$

and of an inference motor which, given an input, allows the system to make progress through the data base until a sensible conclusion is eventually reached.

Neural networks, more especially layered neural networks, can be used as expert systems:

The item A is coded in the first (input) layer and the item B is coded in the last (output) layer. The synaptic efficacies of the system are modified according to the back-propagation algorithm in order to bring the output corresponding to A closer to the desired output B . The system is trained to a whole set of couples of items (A, B) given by human experts, until the synaptics efficacies get stabilized. In the case of an inference of type (Inf. 2), the couples (A, B) and (A, C) are considered as two different couples. It is even possible to have more refined training with probabilistic inferences such as

$$\begin{array}{ll} \text{if } A \text{ then } B & (\text{with } \frac{1}{3} \text{ probability}) \\ \text{or} & \\ \text{if } A \text{ then } B \text{ or } C & (\text{with } \frac{2}{3} \text{ probability}) \end{array}$$

by training the couple (A, C) twice as much as the couple (A, B) . The interesting aspect of these systems is that they are apparently capable of generalization, which means that they are able to give, after training, sensible responses to inputs which do not belong to the training set. This generalization is related to the volume in the space of the interactions which solve the problems. It would appear that the data obey a certain hidden logic that the experts are unable to voice clearly but which the learning rule succeeds in trapping into synaptic efficacies.

There have already been a number of applications for these systems. Most networks used so far are made of three layers, one input layer, one output layer and one layer of hidden units:

- A system for the diagnosis of abdominal diseases has been studied by Le Cun. The inputs are the set of coded symptoms and the outputs the corresponding set of coded diagnoses. The efficiency of the system, the percentage of correct diagnoses, is at least as good as that of human doctors (respectively 70 % and 67 %).
- Neural expert systems have also been applied to the recognition of sonar signals. Here the input signals are the patterns given by a sonar and the outputs are the identifiers of the objects which reflect the ultrasonic waves. The system proves to be as good as humans in distinguishing between rocks and submarines from sonar signals.
- Neural expert systems have been trained to play games, in particular backgammon (Tesauro). The inputs are coded positions of the pieces on the board and outputs are the corresponding coded moves recommended by expert players. Amazingly the system, when confronted with a position it has never seen in the training stage, responds sensibly. In this application the network is made of three layers comprising 50 neurons each. The data base is made of 3200 positions as input items and 22 possible moves as output items. The training stage requires about 50 cycles.

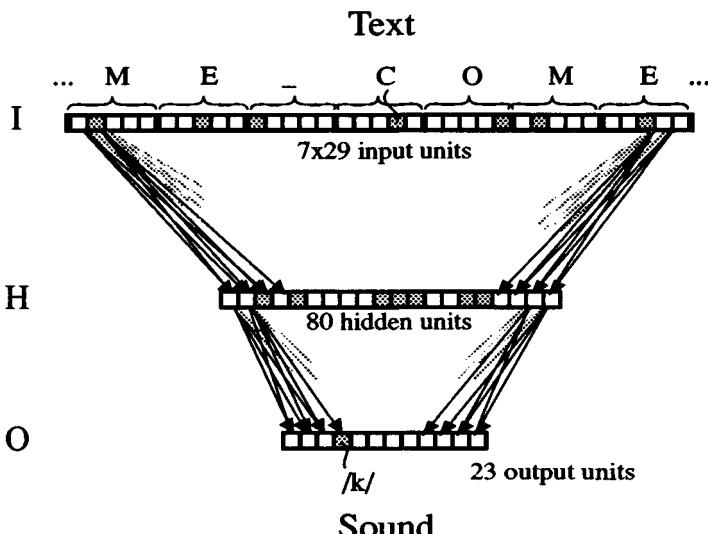


Figure 10.17. The structure of NetTalk, a three-layer feedforward network that learns to speak aloud (After Sejnowski *et al.*).

There have been many other applications. One of the best publicized is NetTalk, a network which learns to read aloud (Sejnowski *et al.*). The network is, as usual, made of three feedforward layers (see Fig. 10.17). The input layer is comprised of 7×29 neurons. Each group of 29 neurons codes for one character. The coding is unary (of the grandmother type with only one neuron firing for one character). The inputs are accordingly strings of seven characters.

- The output layer is made of 23 units, each unit codes for one phoneme.
- The hidden layer comprises 80 neurons.
- The input training set is provided by one page of a simple story which contains about 3000 characters. A phoneme is associated to every string of 7 characters and the system is trained to learn these associations.

It is worth noting that a phoneme is not determined by a single character, not even by a single syllable, but by two or three joint syllables. The system therefore takes context effects into account. 2000 training cycles are necessary for the system to read the text with an 85 % success. The machine is then given a new page for which it has not previously been trained. Obviously the reading quality diminishes but it remains good enough for the authors to claim that the system is in some way able to generalize.

There are many questions regarding these neural expert systems.

- The most fundamental is: what is generalization? How can generalization be measured? When does it occur? A few hints have been discussed at the end of Chapter 9.
- Assuming that an answer can be given to the first question, what is the optimal architecture? How many layers are there? How many neurons are there per layer?

It seems that the systems work best when they comprise three or four layers at most. The number of units of the hidden layer (or of the two hidden layers) must not be too large because the connections then tend to memorize every example without generalizing; it must be large enough however to avoid crude generalizations. In many instances this number is quite small. A system is more likely to generalize if it is made of the smallest possible number of cells and layers, which allows it to utter the right answers for the whole set of training examples. The tiling algorithm of Mézard and Nadal (see section 8.2.2) is precisely devised to build such a minimal layered network and therefore the existence of this algorithm increases hopes that efficiently generalizing machines can actually be assembled. As a rule of thumb, efficient architectures seem to be comprised of three layers with a number of hidden units given by the geometrical average $N_H \simeq \sqrt{N_I N_O}$ of the number of input and output units.

10.6.3 Semantic nets

It has sometimes been observed that the impairment of localized portions of the cortex, spreading over a few square millimeters, is correlated to the loss of very specific pieces of information, for example with the forgetting of the names of vegetables. This has been taken as an indication that concepts are imprinted in well-defined areas of the cortex and that they are related to each other through a network, the semantic net. The concepts make up the nodes of the net. The links mirror the semantic distances between the concepts. Although this idea is suspect to most neurophysiologists and to a great many psychologists as well, it has been used by linguists as a tool for studying the structure of languages and the emergence of meaning and as a way of rendering ambivalent sentences unambiguous through context effects.

Among the various models which have been proposed so far, the approach of Waltz and Pollack is closest to the philosophy of neuronal networks, although they use dedicated, grandmother neurons. For these authors a neuron is attached to every concept. A neuron fires when the concept which it represents is activated. This activity in turn triggers the excitation of neurons, that is of concepts, whose interactions with the firing neuron are excitatory and hinders the excitation of neurons whose interactions are inhibitory.

In semantic nets one distinguishes four types of neurons:

- i) *Input neurons*, one for every word of a language.
- ii) *Syntactic neurons* for the various grammatical types of words: adjectives, verbs, nouns,
- iii) *Lexical neurons* which represent the various meanings of a word.
- iv) *Contextual neurons* whose activities depend on the meaning of preceding sentences.

There are inhibitory synapses between concepts which are mutually exclusive. For example, at the syntactic level, that of parsing, a word is either a noun or an adjective. It cannot be both simultaneously. At the lexical level a word is given one and only one meaning at a time. On the other hand there are excitatory interactions between items which tend to be coupled as nouns and verbs, for example. The interactions are symmetrical and therefore the net is not a layered feedforward system. The dynamics of the net is that of usual neural networks. Owing to the symmetry of interactions the asymptotic behaviors are fixed points which determine the meaning of the sentence. Waltz gives the following ambiguous example (see Fig. 10.18):

‘The astronomer married a star.’

Since the word ‘astronomer’ is semantically associated with celestial bodies, the word ‘star’ first takes its astronomical signification. But after a while, owing to contextual constraints associated with the word ‘married’, the concept of movie-star catches up: the neuron ‘movie-star’, which is first silent, later becomes activated.

This description of neuronal activity is at odds with the concept of collective phenomena which we favour in this text. Networks made of grandmother cells are obviously extremely fragile and the dynamics which has to find routes through a very sparsely connected network is not so parallel. However a simple change of perspective brings both approaches closer together.

First of all a neuron of a semantic net is not necessarily a unique physical neuron, but it can represent a pool of neurons, say a microcolumn or even a larger structure. On the other hand mutually exclusive items can be imagined as steady states of a strongly connected network: as a state of the network settles in one or the other of its fixed points there is a natural exclusion between the items they represent and there is no longer any need for semantic inhibitions. Likewise, two items which attract one another can be thought of as belonging to the same basin of attraction.

The semantic net is really sparse and this can be accounted for only with non-fully connected networks of physical neurons. In actual fact this

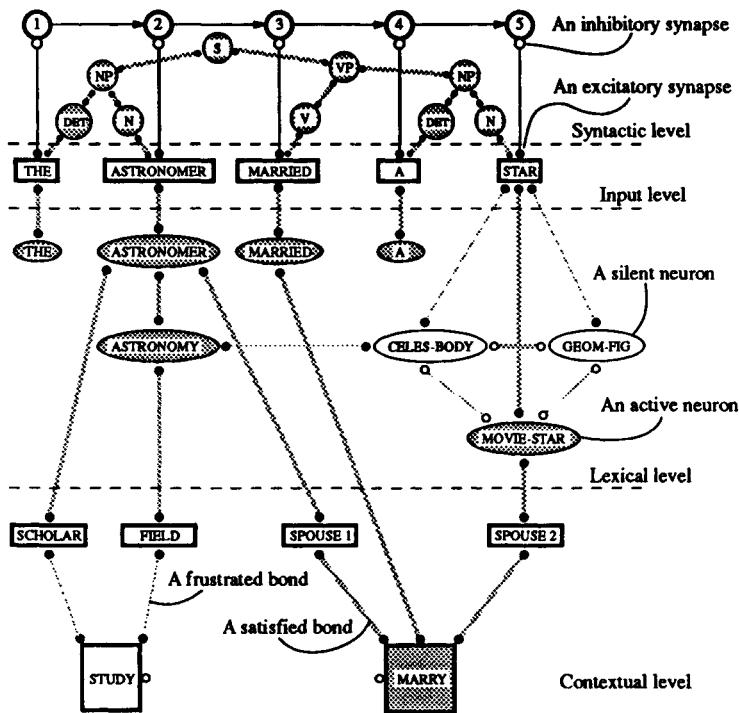


Figure 10.18. A semantic net. Each vertex of the network is a neuron which embodies one of the specific concepts which one encounters in parsing. It is the interplay between the neurons of the net which determines the final meaning of the sentence (After Pollack and Waltz).

fits the topology of biological networks. As we have seen in Chapter 2, real systems are strongly connected at low levels and more sparsely connected at higher levels. The theory of structured networks is now in progress.

NEUROCOMPUTERS

Clearly, any neuronal dynamics can always be implemented in classical computers and therefore we could wonder why it is interesting to build dedicated neuronal machines. The answer is two-fold:

- 1) Owing to the inherent parallelism of neuronal dynamics, the time gained by using dedicated machines rather than conventional ones can be considerable, so making it possible to solve problems which are out of the reach of most powerful serial computers.
- 2) It is perhaps even more important to become aware that dedicated machines compel one to think differently about the problems one has to solve. To program a neurocomputer does not involve building a program and writing a linear series of instructions, step by step. In the process of programming a neurocomputer, one is forced to think more globally in terms of phase space instead, to eventually figure out an energy landscape and to determine an expression for this energy. Z. Pilyshyn made this point clear enough in the following statement (quoted by D. Waltz):

‘What is typically overlooked (when we use a computational system as a cognitive model) is the extent to which the class of algorithms that can even be considered is conditioned by the assumptions we make regarding what basic operations are possible, how they may interact, how operations are sequenced, what data structures are possible and so on. Such assumptions are an intrinsic part of our choice of descriptive formalism.’

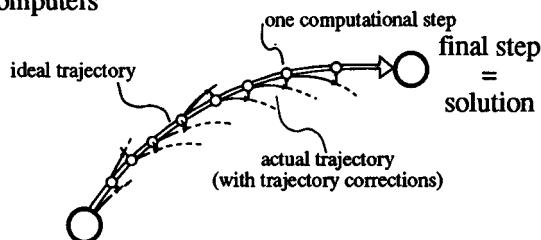
Moreover, neurocomputers are intended to solve problems only approximately, to make analogies and associations. This is in contrast with the classical computers which are designed to keep as much accuracy as possible.

11.1 General principles of neurocomputation

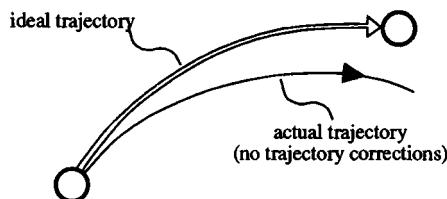
In the forties digital machines competed with analog machines. As it is generally known, the former won. Their victory came from the ability of the digital computer to automatically carry out trajectory corrections of its physical state by ‘binarizing’ the internal state of its components at every step of the computation. Analog machines are not

endowed with such capabilities and the relevance of results they yield depends on the accuracy of the components and on the length of the calculations. Biological computation, because it involves analog devices such as synapses or thresholding mechanisms, is an analog computation and therefore one could imagine that neurocomputers are non-reliable systems. In reality, far from aiming at the final state step by step as in analog computers, the trajectory of the internal state of neurocomputers is trapped in basins and globally attracted by the solution (see Fig. 11.1).

a) Digital computers



b) Analog computers



c) Neurocomputers

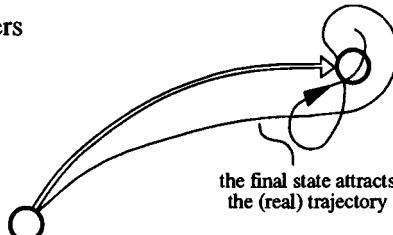


Figure 11.1. Families of computers. a) Digital computers correct the trajectory of the overall state of the machine at every computational step thanks to the 'binarization process'. b) There are no such corrections in analog machines. c) In neurocomputers the state trajectory is trapped by the target solution.

A neurocomputer is a machine which is built to work along the principles of the dynamics of neuronal networks. We have seen that there exist two types of dynamical variables, the neuronal states and the network parameters (the synaptic efficacies and the thresholds). The dynamics of parameters corresponds to learning whereas the dynamics of neuronal states, or relaxation dynamics, corresponds to solving tasks. In fact the dynamics of neuronal states seems to be fairly well established and is worth implementing in hardwired architectures. On the contrary, the general principles driving the learning dynamics are still not as well founded and it is wise to build machines wherein the learning algorithms are not frozen. This approach implies that the computation of the parameters of the network has to be carried out by a powerful host computer or by inboard microprocessors. In this section the main emphasis is therefore on relaxation dynamics.

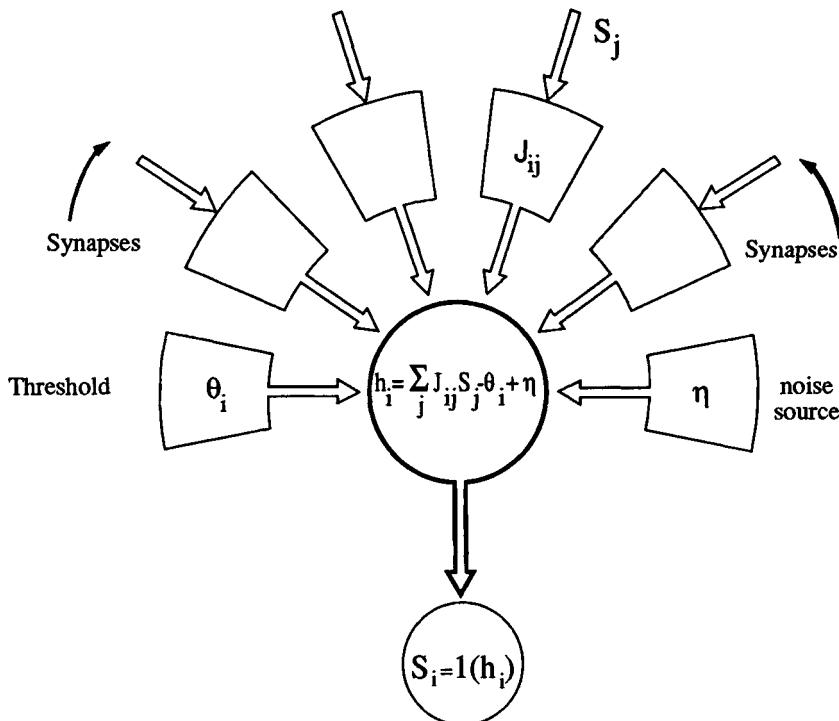


Figure 11.2. Synoptics of neural dynamics.

11.1.1 Classification of machines

Let us recall the basic steps of the neuronal states dynamics (Fig. 11.2). A classical program implementing the neuronal dynamics involves two

nested loops:

- the inner one of length N computes the local field of neuron i ;
- the outer one updates the states of all neurons of the network.

As a whole the computation time scales as N^2 .

A neurocomputer is a machine which aims, in particular, at shortening the computation times. In biological systems the computation of membrane potentials (the local fields) and the updating of neurons are all carried out in parallel and therefore the computation does not depend on the size of the network: the computation times of fully parallel neurocomputers scale as N^0 . It is possible to imagine many

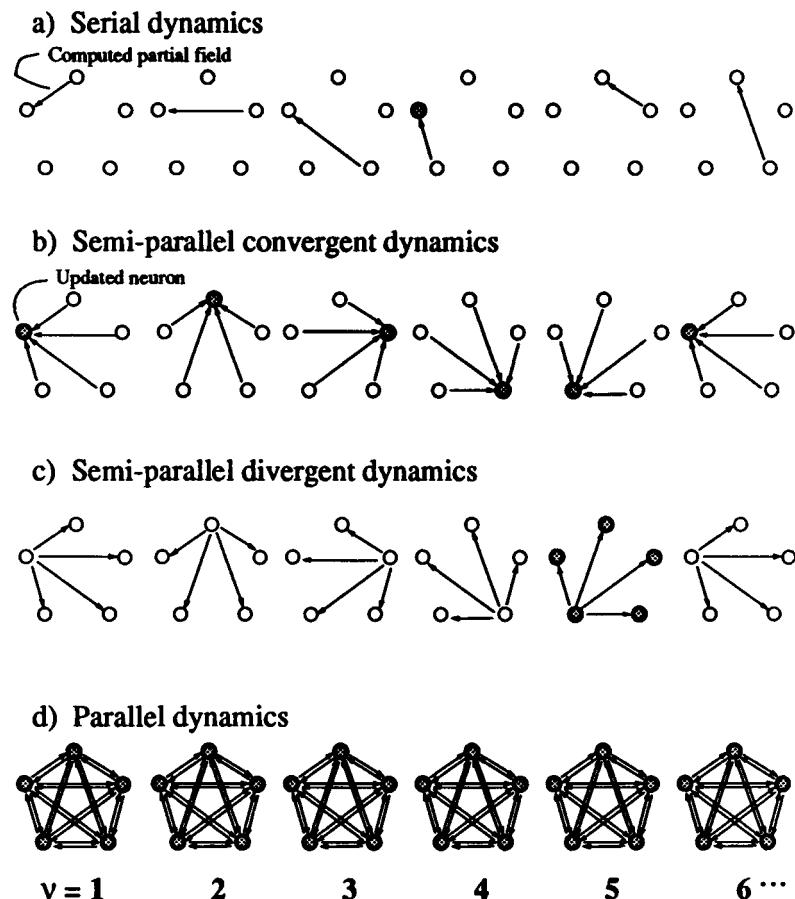


Figure 11.3. Types of dynamics: shaded circles show updated units.

sorts of architectures with various degrees of parallelism (Fig. 11.3). An architecture can be characterized by the way its computation time t_{comp} scales with the size N of the network:

$$t_{\text{comp}} \simeq N^\varphi.$$

- Serial neurocomputers are characterized by $\varphi = 2$ as in ordinary computers. This is the way the components are arranged inside the machine which allows the time gain. As shown in Fig. 11.3.a the contribution of neuron j to neuron i , $J_{ij}\sigma_j$, is computed at time ν , that of neuron $j+1$ to neuron i at time $\nu+1$, etc. The state of neuron i is updated after its field is completed, that is to say after N time steps. The computation then proceeds with another neuron.
- In parallel neurocomputers $\varphi = 0$. All contributions, $J_{ij}\sigma_j$, are computed for all neurons j and all neurons i in one time step (Fig. 11.3.d). These are the most powerful but also the most material-demanding machines.
- $\varphi = 1$ characterizes the semi-parallel machines. Here the computation time increases linearly with the size of the network. We have seen that sequential programs simulating the neuronal dynamics involve two nested loops, each of length N . The first computes the local field,

$$h_i = \sum_j J_{ij} \sigma_j,$$

and the other updates the states according to

$$\sigma_i = \text{sign}(h_i - \theta_i + \eta). \quad (11.1)$$

Semi-parallel neurocomputers are machines which carry out one or the other of the loops in parallel. Accordingly there are two sorts of semi-parallel neurocomputers, those which process the updating of neuronal states in parallel (Fig. 11.3.c) and those which process the updating in series (Fig. 11.3.b). The latter type is called a convergent architecture: all contributions, $J_{ij}\sigma_j$, which make the field on a given neuron i , are computed and added in one time step. The state of i may be updated at once. The former type is called a divergent architecture: the contribution of a given neuron i to the fields of all other neurons is computed in one time step but it is necessary to wait N steps before having all fields computed for all neurons. Then all neurons can be updated simultaneously.

Semi-parallel machines represent a good compromise since the time gain factor is N with respect to serial machines and the gain factor in material and connections is N with respect to fully parallel machines.

- 1) Choose a neuron.
 2) Compute its local field:

$$h_i^0 = \sum_j J_{ij} \sigma_j.$$

- 3) Subtract the thresholds θ_i .
 4) Add a noise η_i , a random number with a given probability distribution, the wider the distribution the more stochastic is the dynamics.
 5) Compare the resulting field

$$h_i = h_i^0 - \theta_i + \eta_i$$

with zero.

- If $h_i > 0$ update the state σ_i of i to $\sigma_i = +1$.
- If $h_i < 0$ update the state σ_i to $\sigma_i = -1$.

- 6) Iterate the process.

The basic steps of the neuronal states dynamics.

This last statement must be qualified, in that whatever the machine the N^2 synaptic efficacies must be stored somewhere.

As a whole, the quality of a neurocomputer design could be defined by the way the product of the time computation by the amount of material scales with the size of the network. If the amount of material is quantified by the number of chips involved in the design, the quality factor is roughly constant, which is another way of saying that ‘Time is money’. As a consequence, the design of a neurocomputer which yields a size-dependent quality factor is probably unsatisfactory. By mixing serial, semi-parallel and parallel architectures it is possible to imagine a quasi-continuum of machines with $1 \leq \varphi \leq 2$.

Remark

The dynamics we have described so far is digital by nature and neurocomputers which implement faithfully its different steps are digital neurocomputers. However, one is often interested in time-averaged activities (which are identical to thermal averages if the system is ergodic). We saw in Chapter 3 that the average activities obey a set of N coupled equations given by

$$\frac{d\langle \sigma_i \rangle}{dt} = -\frac{1}{\tau_r} [\langle \sigma_i \rangle - S(\beta h_i)],$$

with

$$h_i = \sum_j J_{ij} \langle \sigma_j \rangle - \theta_i.$$

It can be profitable to directly embed those equations in hardware electronics. Neurocomputers built along this principle are analog machines. Differential equations are necessarily computed in parallel. Therefore $n = N$ for analog neurocomputers and $\varphi = 0$.

11.1.2 Serial neurocomputers

We have already stressed that the neuronal dynamics algorithm can be implemented in any serial computer (Fig. 11.3.a). A dedicated serial neurocomputer is one which is specially designed to run this algorithm as fast as possible.

Serial neurocomputers come in the form of boards which can be plugged into the bus of PC's. Several products have been developed and marketed so far. The systems are built around a powerful microprocessor which has direct access to a large memory. The memory stores the synaptic efficacies and the neuronal states. Among the products that were commercially available by the end of 1988 one finds the ANZA systems by HNC (Hecht-Nielsen Corporation), the SIGMA system by SAIC, the NEURO-ENGINE system by NEC, the ODYSSEY system by Texas Instruments and the NX series by Human Devices. Let us describe the ANZA system a little more carefully:

The system which Hecht-Nielsen Neurocomputer Corporation first marketed is a board which is to be plugged into a PC host computer. It is built around a 32-bit Motorola microprocessor MC68020. The microprocessor is in close connection with a large dynamic RAM of 4 Mbytes. The synaptic efficacies and the neuronal states are fed into the RAM, which is similar to a look-up table for the microprocessor. The synaptic efficacies are loaded into the microprocessor at a rate of 25 000 connections per second. The system can implement up to about 680 fully connected neurons. More recently, ANZA evolved in a new product called ANZA-PLUS which uses a larger memory of 10 Mbytes. It can be plugged into a PC-AT. This neurocomputer can simulate the dynamics of 1 600 fully connected neurons and is able to compute 1.5×10^6 connections per second.

ANZA and the other neurocomputers we have mentioned above are fully serial machines. Using several microprocessors introduces a small degree of parallelism. This approach has been advocated by Palm. Each microprocessor takes charge of a number of neurons, either in the learning stage or during the processing operations. Let N_P be the number of processors. To take advantage of the multiplicity of processors it is necessary that N_P neurons are simultaneously updated, one at a

time in every processor. Then the computing time scales as

$$\frac{kN^2}{N_P}$$

These are machines which have powerful computational capabilities scattered at the nodes of a network of processors. The COSMIC CUBE is a general-purpose computer built along this philosophy. As far as neurocomputation is concerned, a machine called MARK III, developed by TRW, can be found. It is to be used with a VAX as the host computer and it houses 15 MC68020 microprocessors. IBM is also studying a system called NEP comprising 256 TMS microprocessors in the framework of its PAN project, an acronym for 'parallel associative networks'.

Small microprocessors, called *transputers*, can currently be found on the market. They can process information to and from four other transputers. These transputers can therefore be assembled into lattices. A special language, called *Occam*, has been developed for the transputers. It is tempting to use these devices as building blocks of a neural network. The transputers naturally assemble themselves into square planar lattices. But they can also be hierarchically organized, one transputer being connected to three others, each connected to three others and so on. The problem, nevertheless, is that of routing: the state of one neuron implemented in one transputer must be sent to other neurons through whole chains of transputers. All these flowing informations compete at the entries of the computers, the result of which is a number of access conflicts which are not easily solved. Actually, the most efficient architecture, as far as routing times are concerned, is a set of randomly connected transputers. Indeed, the average number of steps between two transputers in a crossbar architecture scales as $(N_T)^{1/2}$, where N_T is the number of transputers, and as $\log(N_T)$ in random networks (Wallace).

11.1.3 Electronic parallel neurocomputers

Parallel neurocomputers are machines which materialize all the components of neural networks, neuronal states, synapses, thresholds and noise. We saw that the convergence time for these machines does not depend on the size of the network (Fig. 11.3.d). It can easily be of the order of one microsecond or less. This is several orders of magnitude shorter than the fastest serial machines. If one wants to build general-purpose machines the price to be paid is a tremendous inflation of the material which is necessary for their construction.

a) The structure of the network

The most standard way of realizing such a machine is to appeal to the

classical devices of digital electronics. A neuron is a small ALU (arithmetic and logical unit) processor carrying out the additions, multiplications and comparisons involved in neural states dynamics. The synapses are endowed with computational capabilities which they use in the learning stage. This approach is not realistic. It leads rapidly to prohibitive numbers of transistors, even for small networks. A chip comprising 22 neurons with $+1, 0, -1$ programmable synapses has been built at Caltech by Sivilotti. The approach seems to have been later abandoned.

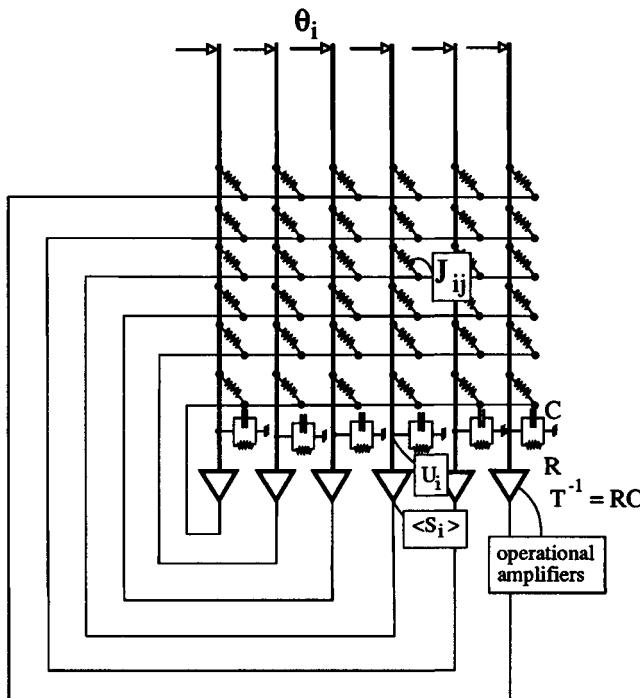


Figure 11.4. The parallel architecture of analog neurocomputers put forward by Hopfield.

Hopfield suggests the use of machines comprising analog devices such as operational amplifiers as opposed to using binary neurons. The currents given by the amplifiers represent the time averages of activities rather than the instant activities themselves. The synapses are mere resistors. Let us look at Fig. 11.4. Let

- I_i be the current which is injected into the input line of neuron i : it represents the thresholds, the external field and the noise field.

- U_i be the input potential of amplifier i .
- V_i be the output potential of amplifier i . V_i is the (analog) state of neuron i .
- $J_{ij} = 1/R_{ij}$ be the conductance associated with the connection (ij) .
- C and R be the capacity and the resistance representing the load impedance of the amplifier.

Then the current conservation at the entry of the amplifiers yields

$$\sum_j J_{ij} V_j + I_i = \frac{U_i}{R} + C \frac{dU_i}{dt},$$

with $V_i = \mathcal{S}(U_i)$, where $\mathcal{S}(x)$ is the characteristics of the amplifiers.

We recognize the equation determining the dynamics of average neuronal activities (see section 3.3.5). Neuronal networks, however, need to have negative as well as positive J_{ij} 's whereas the resistors are only positive quantities. The way out of the problem is to double the number of amplifiers and the number of lines: one associates the amplifiers

$$V_i^+ = \mathcal{S}(U_i) \quad \text{and} \quad V_i^- = -\mathcal{S}(U_i)$$

to every neuron i . A synapse is represented by two resistors among which one is set to zero according to the sign of the synapse (see Fig. 11.5).

b) Possible realizations of the matrix of connections

If the machine has always to solve the same problem, that is if it is not necessary to modify the connections, the values of the resistors can be frozen once and for all.

For example, the resistor states can be binary: either the contact between the ‘axon’ j and the ‘dendrite’ i is ‘on’ ($J_{ij} = 1$) or it is ‘off’ ($J_{ij} = 0$). Taking the two sorts of lines into account, the synaptic contact can actually take three values, $+1$, 0 and -1 . The main interest of frozen synapses is that they are liable to be strongly integrated on silicon substrates: a layer of amorphous silicon is sandwiched between two sets of metallic wires perpendicular to each other. One set is the set of axons j and the other the set of dendrites i . They are less than one micron apart. All the initial efficacies are $J_{ij} = 0$. To change a particular connection from $J_{ij} = 0$ to $J_{ij} = 1$, currents are injected in lines i and j which are strong enough to make the amorphous silicon melt in between the two wires. A 512×512 matrix has thus been made at Bell Laboratories.

A programmable machine is obviously more interesting. Modifiable binary synapses can be materialized by special devices such as floating grid transistors similar to the ones used in the E² PROM’s. These

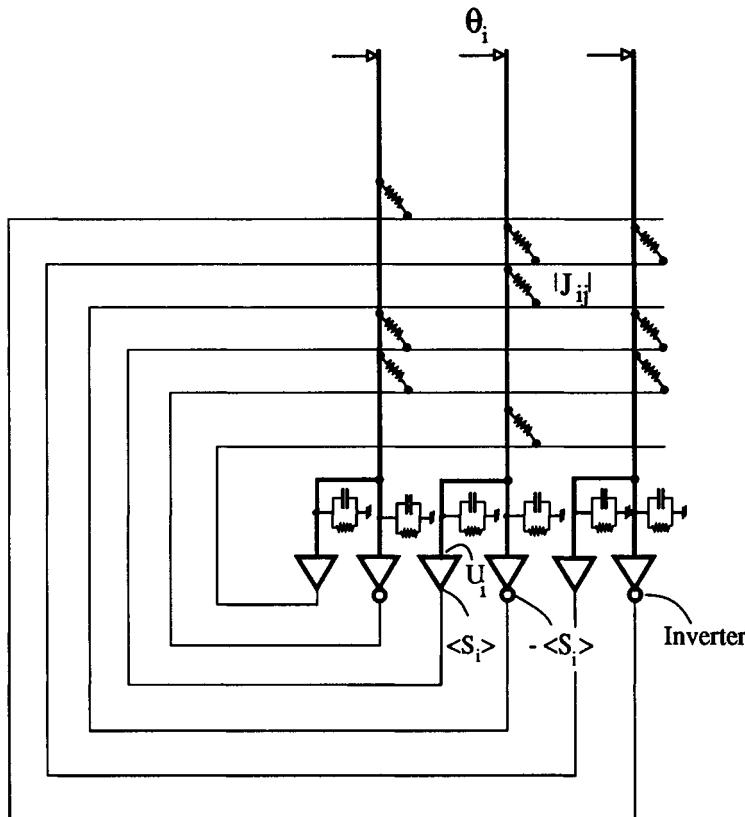


Figure 11.5. How to implement synaptic efficacies of both signs in analog parallel neurocomputers.

devices are actually switches, the states of which can be controlled by external lines. A system comprising 54 neurons has been built at Bell Laboratories and a chip of 280 neurons at Caltech. JPL has also built 32×32 binary synaptic matrix CMOS chips. The chip is cascadable.

There exist materials displaying irreversible electrical properties which could be useful as supports of synaptic efficacies. For example, deposits of metallic vapors in a device called a memistor have been used by Widrow in Adaline. It could be also interesting to use thin layers of amorphous hydrogenated silicon $\alpha\text{-SiH}$ or of non-stoichiometric manganese oxide MnO_{2-x} , which displays irreversible characteristics. Thomson in France studied the possibility of making reversible memory matrices using pyro-electric materials. At the MIT Lincoln Laboratory J. Sage proposes the use of MNOS/CCD techniques.

11.1.4 Optical parallel neurocomputers

It has been suggested that optical devices, instead of electronic ones, could be convenient materials with which to build neurocomputers.

In particular, the transportation of information by light beams solves the difficult problem of routing that appears in the design of VLSI chips owing to their essentially two-dimensional structures. Another advantage of using optical components is the possibility of storing a considerably large amount of information on cheap pieces of material. The disadvantage is that it has been difficult (at least up to now) to integrate the optical devices on any large scale and that therefore these machines would not be as fast as would have been expected. Two sorts of systems have been proposed:

a) *Information embedded into an optical mask (Fahrat and Psaltis)*

The mask is an array of $N \times N$ optical filters whose transparencies are proportional to the synaptic efficacies J (see Fig. 11.6). The neurons are materialized by a set of N photodiodes which can be 'on' ($S_j = 1$) or 'off' ($S_j = 0$). Thanks to optical fibers the light emitted by a diode j lights

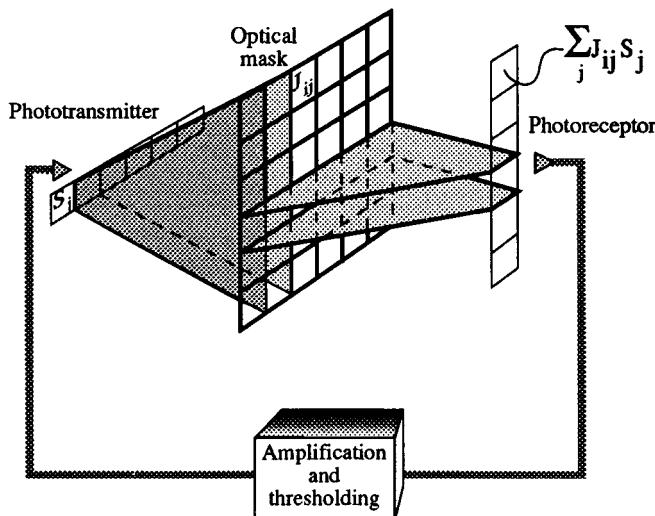


Figure 11.6. An optical implementation of analog parallel neurocomputers based upon the use of optical masks.

up a row of filters in the mask. The output intensity of the filter (ij) is therefore $J_{ij}S_j$. Other sets of optic fibers convey the various outputs of a line of filters of the array on a photoreceptor i . The light intensity

collected by a receptor is therefore

$$\sum_j J_{ij} S_j.$$

All the output signals produced by the receptors are thresholded, possibly blurred by some noise sources and recycled in the photoemitters. N. Fahrat suggests using the speckle phenomenon as an optical noise source.

b) *Information embedded in holograms*

Let us assume that a photographic plate is lit with two coherent beams coming from two phase objects A and B . The light intensity on the plate which determines the opacity ΔT of the plate at point \mathbf{r} is given by

$$\Delta T(\mathbf{r}) = k [F_A(\mathbf{r}) + F_B(\mathbf{r})] [F_A(\mathbf{r}) + F_B(\mathbf{r})]^*.$$

The plate is now lit with the coherent light beam of object A alone. The complex amplitude of the output signal is

$$\begin{aligned} F(\mathbf{r}) &= (1 - \Delta T) F_A(\mathbf{r}) \\ &= (1 - k) F_A (|F_A|^2 + |F_B|^2) - k (F_A F_B^* F_A + F_B |F_A|^2). \end{aligned}$$

Since $|F_A|^2 = F_A F_A^* = |F_B|^2 = 1$, the output amplitude is given by

$$F(\mathbf{r}) = (1 - 2k) F_A - k F_B + \text{noise}.$$

Therefore the image of A tends to disappear, whereas that of B appears: the object A evokes the object B , as in associative memory. In the very same way it is possible to associate several pairs of patterns. More information as regards the holographic approach of neurocomputation may be found in Kohonen's book on associative memory.

On the other hand, it has been suggested that the information could be stored in the form of three-dimensional holograms. As the information density would then be of the order of 1 bit/ μm^3 , the memory storage capacity would be enormous. There exist materials, photorefractive crystals, whose refraction index can be modified irreversibly by sufficiently powerful light beams. LiNbO_3 is such a material. Unfortunately, the light intensity that is needed for the imprinting process to be efficient is so high that the crystal often cracks.

11.2 Semi-parallel neurocomputers

11.2.1 A classification of semi-parallel neurocomputers

For each type of semi-parallel architecture, the semi-parallel divergent and the semi-parallel convergent designs that we have introduced in

section 11.1.1, there are two ways of computing the local fields, according to whether one appeals to systolic (or pipeline) computations or not. Therefore the semi-parallel machines are classified along four types of architectures (see Table 11.1). The systolic computations are carried out by several processors arranged along a line. The input of a processor is the output of its left neighbor, for example, and the result of the computation is sent to its right neighbor in much the same way as industrial products are manufactured along assembly lines. Here the products are the local fields and a neuron i is updated as soon as the computation of its local field is completed. The dynamics of the computation of local fields is depicted in Fig. 11.7 for each type of neurocomputer.

Dynamics	Type	Characteristics
Updating states in parallel	I	Systolic field calculation
	II	Serial field calculation
Updating states in series	III	Pipe-line field calculation
	IV	Parallel field calculation

Table 11.1.

11.2.2 Describing the architectures of semi-parallel neurocomputers

a) The serial states updating semi-parallel architectures

These machines are built around shift registers which store the states of the system. They are made of N units which perform the elementary operations of the neuronal dynamics in parallel and store the result of calculations. Each unit i is comprised of a memory which stores the connections J_{ij} , $j = 1, 2, \dots, N$, a register for the field, one for the threshold, possibly a memory to store a noise distribution, an adder, a multiplier and a comparator which is made active by a gate δ .

- The type I (systolic) design. — In this design the units have access to the states of the shift register they are facing (see Fig. 11.8). At time ν the field h_i of unit i transforms according to

$$h_i \mapsto h_i + J_{ij} \sigma_j,$$

where $j = (\nu + i) \bmod (N)$.

After a complete turn, the threshold and the noise contributions are added and the fields are compared with zero. The comparator gates are then open and the new states are fed in parallel in the shift register.

$$\begin{aligned}
 h_1 &= J_{11}\sigma_1 + J_{12}\sigma_2 + J_{13}\sigma_3 \\
 h_2 &= J_{21}\sigma_1 + J_{22}\sigma_2 + J_{23}\sigma_3 \\
 h_3 &= J_{31}\sigma_1 + J_{32}\sigma_2 + J_{33}\sigma_3
 \end{aligned} = \boxed{\begin{array}{ccc} \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \end{array}} \quad \begin{array}{l} \circ \text{ neuron 1} \\ \circ \text{ neuron 2} \\ \circ \text{ neuron 3} \end{array}$$

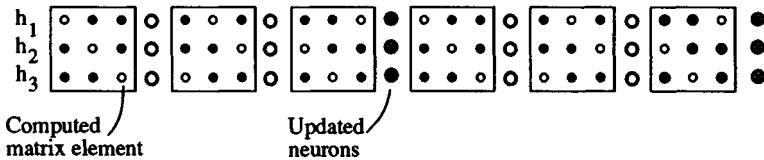
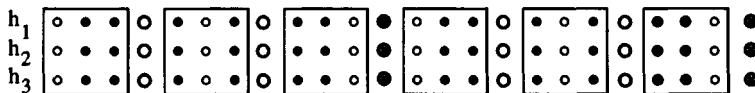
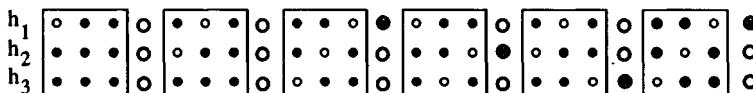
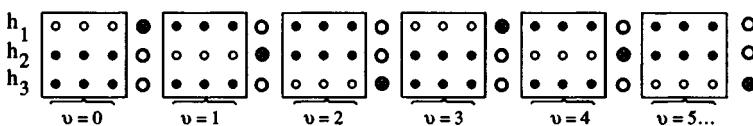
Type I Parallel updating Systolic field calculation**Type II** Parallel updating Serial field calculation**Type III** Serial updating Pipeline field calculation**Type IV** Serial updating Parallel field calculation

Figure 11.7. The four types of semi-parallel neurocomputer architectures in a fully connected three-neuron network. Small open circles show computed elements. Large filled circles are updated neurons.

Afterwards the contents of the field registers are reset to zero and the system is ready for a new cycle.

This design has been put forward by M. Weinfeld. In this approach the operations are carried out by an ALU and the effect of noise is simulated by directly disturbing the states in the shift register.

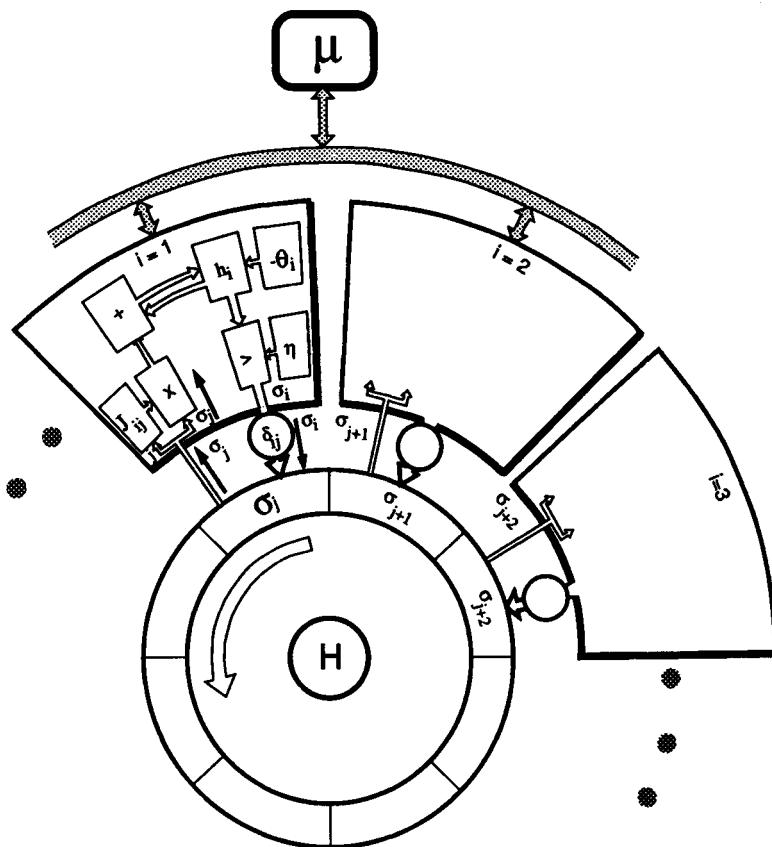


Figure 11.8. Type I semi-parallel systolic architecture (After Weinfeld).

- The type II design. — The components of this machine are the same as in the previous one. The difference lies in the fact that the very same state, that stored in a position of the shift register defined by a pointer, is fed through a bus line to all units of the computer (see Fig. 11.9). The increase in the field at time ν is given by

$$h_i \mapsto h_i + J_{ij}\sigma_j, \quad \text{with } j = \nu \pmod{N}.$$

b) The parallel states updating semi-parallel architectures

These systems are made of units j which contain a storage memory for connections J_{ij} , $i = 1, 2, \dots, N$ (note the swapping of the role of indices i and j with respect to the serial semi-parallel architectures), a register for state j , a multiplier and a gate δ for updating. The whole set

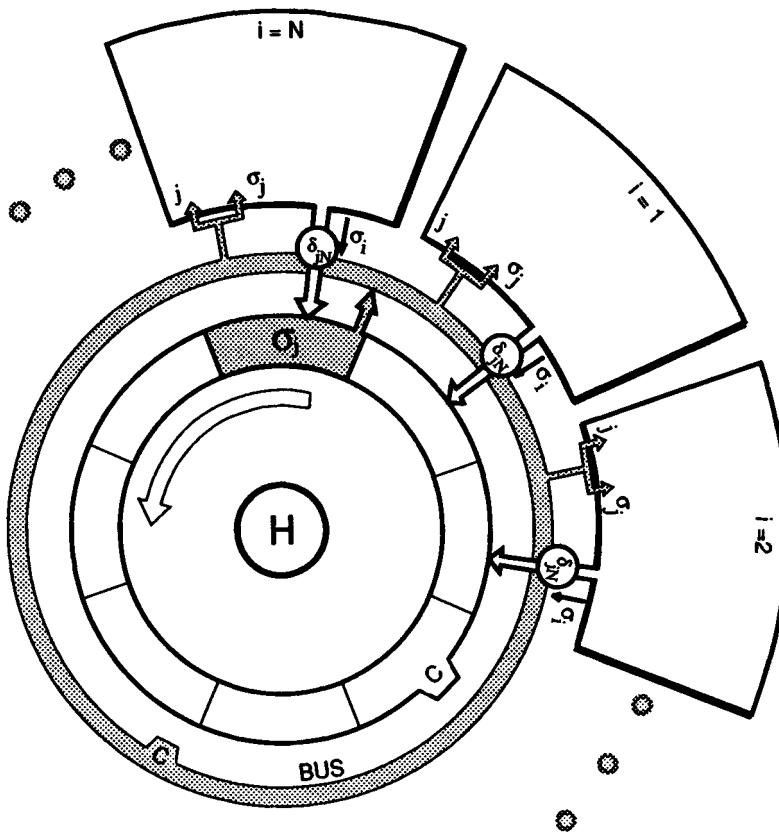


Figure 11.9. A type II design which combines a parallel state updating and a serial field computation.

of thresholds and the noise distribution are stored in two independent memories.

- The type III (pipeline) design. — The pipeline is a string of adders and registers which store the partial fields (see Fig. 11.10). At time ν the field stored in register j (that facing unit j) is increased according to

$$h_i \mapsto h_i + J_{ij} \sigma_j, \quad \text{with } i = (\nu - j) \bmod (N).$$

The thresholds and the noise are added in the first stage of the line and one unit, that of index $i = \bmod (N)$, is updated at every step.

- The type IV design. — This is the most simple architecture one can conceive: all contributions $J_{ij} \sigma_j$ of the field of a given neuron i are computed in parallel and the state of i is updated accordingly (see Fig. 11.11). The problem involves that of carrying out the parallel

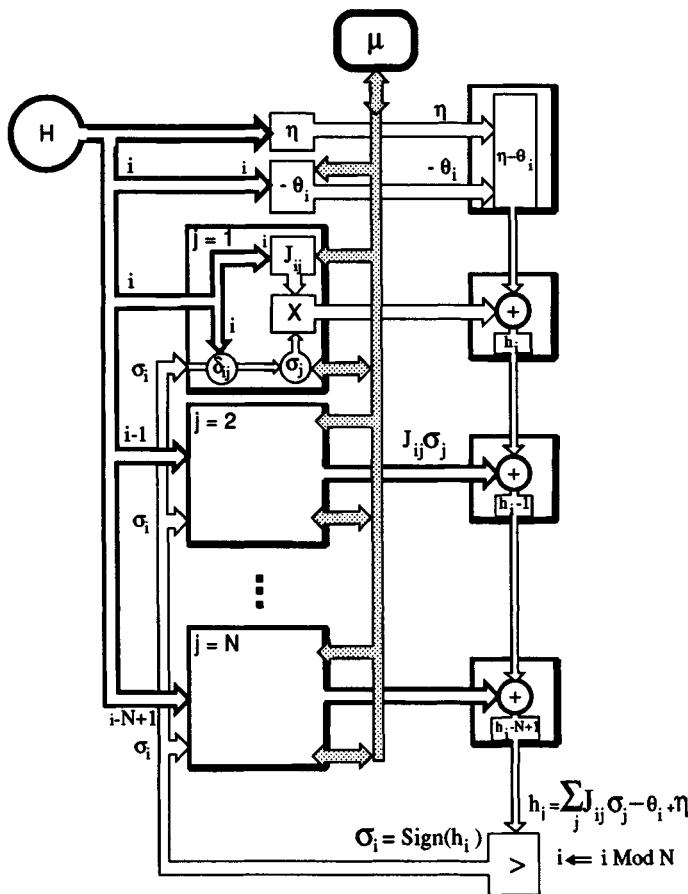


Figure 11.10. Type III semi-parallel architecture. The field computation is pipelined and the neuronal states are updated in series.

addition. There are two solutions: one appeals either to an analog adder or to parallel digital adders called Wallace trees (Fig. 11.12). The first approach has been implemented in several machines, in particular in MIND-128, a neurocomputer built at the CEA in 1986 (Van Zurk, Mougin and Peretto). This machine is more carefully described below.

c) MIND-128: a semi-parallel asynchronous neurocomputer

MIND is an acronym for ‘machine implementing neural devices’. In this machine the neuronal states are binary, the synaptic efficacies and the thresholds are stored in RAM memories (see Fig. 11.13). The parallel computation of all local fields is carried out analogically by the

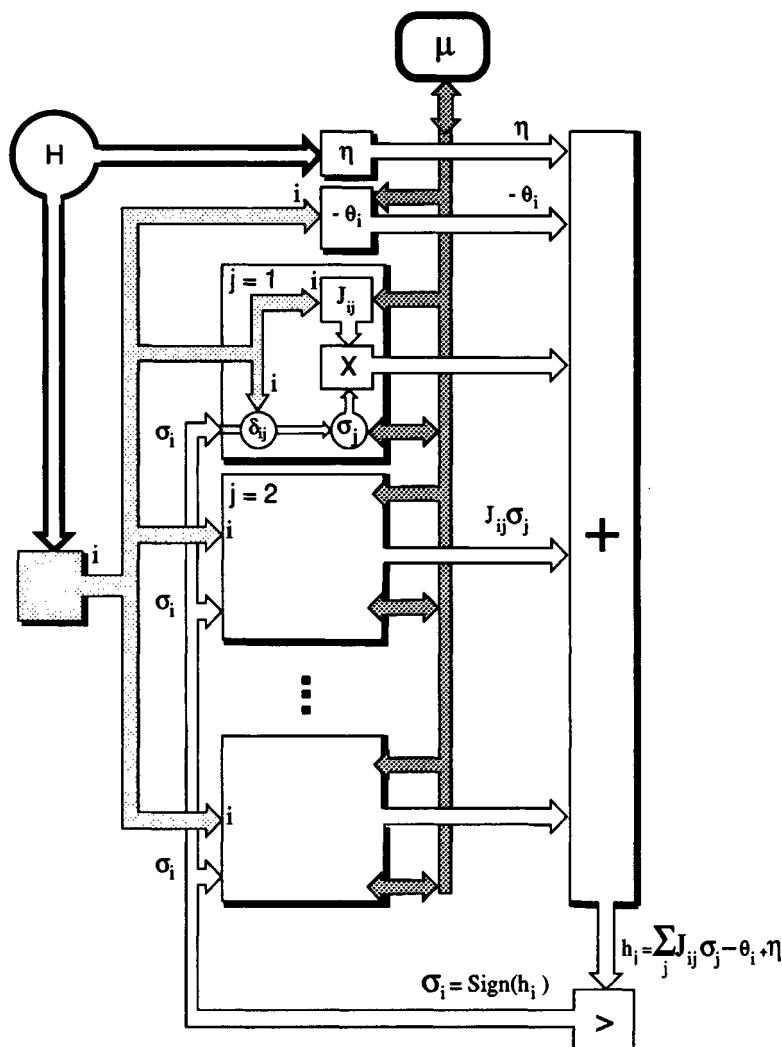


Figure 11.11. Type IV neurocomputers. Parallelism is used to compute the fields and the neuronal states are updated in series.

summation of currents and the state of the active neuron is determined by an analog comparator. The noise signals are stored in an E^2 PROM memory which is filled according to the noise distribution which has been chosen. Likewise, the string of labels of neurons to be updated is memorized into another E^2 PROM.

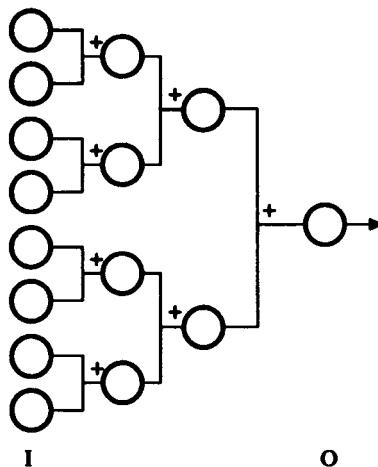


Figure 11.12. A Wallace tree adder.

The dynamics of this neurocomputer proceeds along the following steps:

- 1) The host computer calculates the $N = 128$ thresholds and the $N^2 = 16384$ synaptic efficacies associated with a given problem. These values are loaded into the corresponding memories through the ports of the host computer. A synaptic memory of 256×4 bits is associated with every neuron j . It contains two pages. The left page is devoted to the efficacies J_{ij} and the right page to the opposite efficacies $-J_{ij}$. This trick avoids carrying out explicit multiplications $J_{ij} \sigma_j$ by associating a page with a neuronal state.
- 2) The host computer sets the neuronal states to their initial values.
- 3) The system comprises a clock. The first tick selects the first neuronal label, i for example. The synaptic efficacies, J_{ij} if $\sigma_j = 1$, $-J_{ij}$ if $\sigma_j = -1$, are simultaneously loaded in the DAC's (digital to analog converters). The threshold θ_i is likewise loaded in the threshold DAC and the first noise contribution η .
- 4) All currents from the various DAC's add up in the bipolar charge line.
- 5) The current determines the output state of a comparator. This output is used to reupdate the state of the neuron i .
- 6) The following tick selects another neuron, and so on.

The synaptic efficacy range is 4 bits, that of thresholds is 8 bits. To make the electronic design as simple as possible, RAM memories with separated input and output lines are chosen. 8 bit-DAC's are also used. The synaptic efficacies feed the 4 most significant bits of the DAC's. In

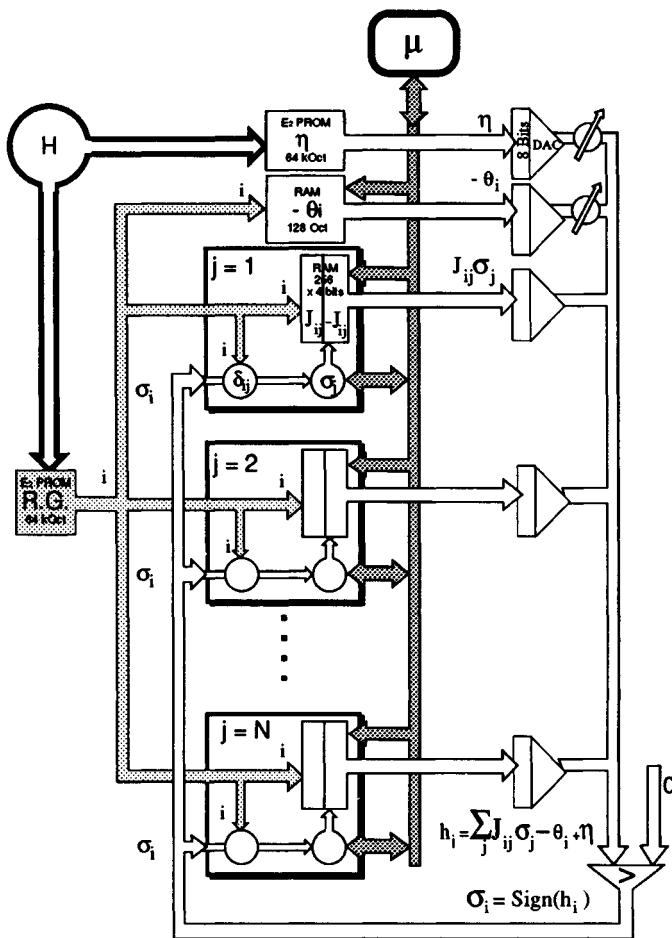


Figure 11.13. MIND-128: an analog type IV neurocomputer comprised of 128 units. The computation of the local fields is carried out by using current summations.

that way the accuracy of the total current on the line is better than half of the change owing to the modification of one single less significant bit of efficacies. The system is built on a series of boards. There are 16 boards housing 8 neurons each, one board for the comparator, one board for the noise, one for the thresholds and one for the interface with the host computer; that is, 20 boards plugged into a VME bus line. To minimize the fluctuations due to the current sources every board is given

its own current source. One of the advantages of this architecture is that the source associated with the noise DAC can be made adjustable, thus giving a very practical means of controlling the noise amplitude (the temperature of the network). The thermal annealing process can be carried out 'by hand'. The states of neurons are made directly visible on an array of 8×16 diodes.

This neurocomputer works quite satisfactorily. It solves all the problems of associative memory, of combinatorial optimization and of solid-state physics (two-dimensional magnetism) which it is programmed for.

The machine is not designed to optimize its computational speed, however. Its clock frequency is 150 kHz only, which is really slow when it is compared with the 20-MHz clocks of modern microcomputers. The speed is actually limited by the memories we use. Even so the machine has proved to be three times faster than the fastest serial machine one can find on the market, the Cray-XMP. The convergence times measured on a benchmark associative memory problem are 3 ms for the system and 8 ms for the Cray-XMP using the same fully vectorized algorithm (and look-up tables to generate the random numbers when necessary). The consumption of the machine is 20 watts.

A 32-neuron machine made at JPL by Thakoor, Moopenn, Lambe and Khanna in 1987 and IRENE, a neurocomputer comprising 1024 units made by the Thomson firm in 1988, have also been built using this architecture.

The reliability of semiparallel analog machines is limited by the accuracy of the analog components they are made of, that of the DAC's in particular. The accuracy of a DAC is of the order of its less significant bit and the current fluctuations, owing to the DAC's in the current summation line is therefore of the order of $N^{1/2}$. This must be smaller than the change brought about by the smallest variation of one synaptic efficacy. Let b_D be the number of bits of a DAC, b_J the number of irrelevant bits (which means that the synaptic efficacies are coded on the $(b_D - b_J)$ most significant bits of the DAC's) and $N = 2^n$ the number of neurons. Then one must have

$$b_J > \frac{1}{2}n \quad \text{and} \quad b_D - b_J > b_E,$$

where $b_E = kn$ is the number of bits which are necessary for the neurocomputer to tackle the problems it has to solve. The maximum size of a network is then given by

$$n = \frac{b_D}{k + \frac{1}{2}} = \log_2(N).$$

There exist 14-bit DAC's and even DAC's with higher numbers of bits. However, more accurate devices reach their equilibrium so slowly that

the time gain due to parallelization is lost. With $k = 1/2$ and $b_D = 14$, 4096-unit analog neurocomputers would be feasible and with $k = 1$ the size would be limited to 1024 neurons. $k = 1$ is the value which is necessary to program combinatorial optimization problems in a neurocomputer. In reality there are many other sources of noise in those machines which limit the size to even lower values. 512 neurons seems to be the limit size of general-purpose analog neurocomputers. For associative memory only one-bit J_{ij} s are needed, since clipping the synaptic efficacies only reduces the memory storage capacity by a factor of 2 or so. Then $k = 0$ and $N \simeq 16 \times 10^6$ neurons: there is no practical limitation to the size of the network in that case.

11.2.3 Possible trends in neurocomputer architectures

Depending on the envisaged applications and the current status of technology, various families of neurocomputers are likely to be developed.

a) There are applications which do not demand precise synaptic strengths. Most problems that are solved by biological systems are probably of that kind. The components of the machine can be crudely defined with synapses ranging on a few bits. Analog devices may also be used. The lack of accuracy makes the components cheap to produce in massive quantities by using the techniques of very large integration on silicon (VLSI) for example. In this approach the natural trend is to make fully connected networks whose size is as large as technology permits.

b) To solve problems of all the types we reviewed in Chapter 10 it is necessary, however, to have general-purpose neurocomputers. These machines need accurate components, the number of which increases very fast when the size of the network becomes larger and larger. To avoid this difficulty one must appeal to semi-parallel architectures. We have seen that semi-parallel implementations of networks of size N run N times faster than conventional machines. This factor could be even larger, the extra factor coming from the necessity for a serial machine to access remote memories and to spend some tens of ticks carrying out the arithmetical operations involved in the computations of the local fields. This time may be saved in hardwired machines. It is to be regretted that the semi-parallel machines loose the same factor of N when they are compared with fully parallel neurocomputers. Nevertheless the advantages overcompensate the disadvantages: on the one hand the amount of necessary material is considerably reduced even though the reduction factor is not quite N since, as already stressed, the N^2 synaptic efficacies have to be memorized somewhere. On the other an interesting consequence resulting from this saving is that the design of semi-parallel architectures does not raise any routing problems.

c) The size of machines, be they of either type, will probably be limited in the foreseeable future to a few thousand neurons. This is very far from what is observed in natural nervous central systems. However, even these natural systems are not fully connected: modularity seems to be the rule. It is therefore tempting to imagine that the next generation of neurocomputers will be made of a number of *modules*. The connectivity of the modules could be inspired by that of cortical columns with as many intra-modular connections as inter-modular connections. The type IV semi-parallel architecture is well suited to that design. The constraint, which is specific to this type of architecture, is that it is compulsory for a neuron of a given module to share its axonal connections between two modules exclusively, namely its own module and another well-defined module. This connectivity is very reminiscent of that of cortical columns.

d) The place of optical techniques in the building of neurocomputers is not clear yet. Optical integration is far from yielding devices as compact as those one obtains using classical techniques of integration on silicon. Moreover the technique is power-demanding, which could result in difficult energy dissipation problems. On the other hand optics is well suited to bringing signals from one point to another. It is therefore likely that future neuronal architectures will be hybrid in nature with signals processed by integrated silicon circuits and transmitted through optic fibers. For example it is necessary that each module of a modular neurocomputer knows the states of all neurons at any moment. This information could be brought to all modules through an optic fiber bundle, with every fiber embedding the state of one neuron.

12

A CRITICAL VIEW OF THE MODELING OF NEURAL NETWORKS

This text started with a description of the organization of the human central nervous system and it ends with a description of the architecture of neurocomputers. An unattentive reader would conclude that the latter is an implementation of the former, which obviously cannot be true. The only claim is that a small but significant step towards the understanding of processes of cognition has been carried out in recent years. The most important issue is probably that recent advances have made more and more conspicuous the fact that real neural networks can be treated as physical systems. Theories can be built and predictions can be compared with experimental observations. This methodology takes the neurosciences at large closer and closer to the classical ‘hard’ sciences such as physics or chemistry. The text strives to explain some of progress in the domain and we have seen how productive the imagination of theoreticians is.

For some biologists, however, the time of theorizing about neural nets has not come yet owing to our current lack of knowledge in the field. The question is: are the models we have introduced in the text really biologically relevant? This is the issue I would like to address in this last chapter. Many considerations are inspired by the remarks which G. Toulouse gathered in the concluding address he gave at the Bat-Sheva seminar held in Jerusalem in May 1988.

12.1 Information structures the biological system

All theories of natural phenomena necessarily simplify the real objects they claim to explain. This is as true in physics as it is in biology. Happily for physicists, it happens that a classification of physical systems naturally arises from the existence of relatively well-defined scales of energy, length and time. Since there are no gaps between the various scales that are involved in living systems, a classification based upon such physical criteria is not possible in biology. This is not to say that

there is no way out other than that of studying a biological system as a whole, but that the nature of criteria allowing for a classification are different in physics and in biology. Clearly, *information* is the relevant criterion in biology: different types of information are associated with different types of biological structures, even though the structures may be closely intertwined in the organism. A specific code is attached to every system. For example, it is possible to distinguish at least three types of codes, the genetic code, the immune code and the neural code. Probably other codes also exist such as a hormonal code and so on. As a first approximation it is possible to focus attention on the properties of a specific system while ignoring the other systems, just as chemists are not really concerned with the nuclear forces of the atoms they combine. The main properties of the neural system are embedded in the neural code and it is on the neural code that attention must be primarily focused. As far as the aim is to understand the neuronal activity, the study of the genetic code is of secondary importance.

12.2 The neural code

Up to now the neural code had been defined implicitly: the neural activity is fully determined by the strings of action potentials which the neurons emit. All neurons are treated on an equal footing. The way the neurons are connected is important, but the nature of spikes is the same whatever the neuron they originate from. There is no information embedded in the shapes of spikes, and the relevant quantities are the instantaneous frequencies which are defined as the inverse of interspike intervals. The precise time cross-correlations between spikes emitted by different neurons do not matter. Interspike frequency is the most widely accepted type of coding and it is true that raster recordings of impulse discharges of a set of cells show patterns of activities which strongly depend on the specificities of stimuli. The patterns are characterized by the average frequencies of the cells of the set. The frequency is high for some cells of the set. It is low for others.

There exist however neurons which do not spike, mainly the sensory neurons, the photoreceptors for example. Their outputs are essentially analog signals. This does not jeopardize the theory, since the sensory neurons are generally connected with deeper neurons which transform the analog signals into trains of spikes; the stronger the input, the higher the frequency of the trains. There also exist electrical synapses or gap junctions, which transmit damped but non-processed signals from one neuron to the other.

However, another, basically different, type of neural coding has been proposed by M. Abeles which he calls synfire chains: according to Abeles

the activity of neural networks manifests itself by the synchronous firing of all the neurons in a group of neurons which in turn triggers the firing of an other group and so on. Here synchronicity is essential and information is embedded in time cross-correlations between neural activities. The synfire chains are more carefully described in the next section. As a matter of fact, the theories we have exposed ignore the existence of slow, coherent activities altogether. This is the case of the theta waves of the hippocampus, for example. Occasionally one also observes bursting activities. The roles of those activities is not clear. They are ignored in the present models, but it could well be that their influences have been underestimated.

The neuronal activity can also take the form of a set of oscillators. This is the case of the olfactory bulb, where specific oscillators are excited by given types of molecules which the atmosphere carries. Recently, it has been suggested that information could be coded in the relative phases of these oscillators.

From these remarks it may be concluded that one must be cautious about the relevance of the models we have developed so far as a universal framework for the explanation of all biological neural processes.

12.3 The synfire chains

A single microelectrode is able to record the spiking activities of several neurons simultaneously. In effect the electrical field emitted by a neuron depends on the relative position of the neuron with respect to that of the electrode, and each neuron in the vicinity of the electrode gives rise to a specific signal. A computer analysis can disentangle every contribution to the overall activity by using template matching techniques for example. The activities of up to five or six neurons can thus be recorded simultaneously. M. Abeles applies this experimental procedure to the auditory cortex of cats. He observes striking systematic time coincidences between the firing times of some of the recorded neurons. The delays may be of the order of several hundred milliseconds whereas the dispersion of delays is as low a few milliseconds. For Abeles the explanation lies in the fact that neurons of specific sets of cells tend to fire simultaneously, thus triggering the activity of another set, and so on. He considers neural networks with partial connectivities and looks for the probability that a set of fully connected neurons makes full contact with at least another set of neurons of same size. For example, a network consists of 20000 neurons, with each neuron connected to 5000 neurons (the connectivity is 0.25). If a set of 5 neurons is randomly chosen there is a probability close to unity that another set of 5 neurons can be found in the network which receive synapses from all neurons

of the first set. If one starts from a set of 7 neurons the probability of finding another set of 7 neurons receiving complete connections from the first group of neurons is close to zero (see Fig. 12.1). The transition

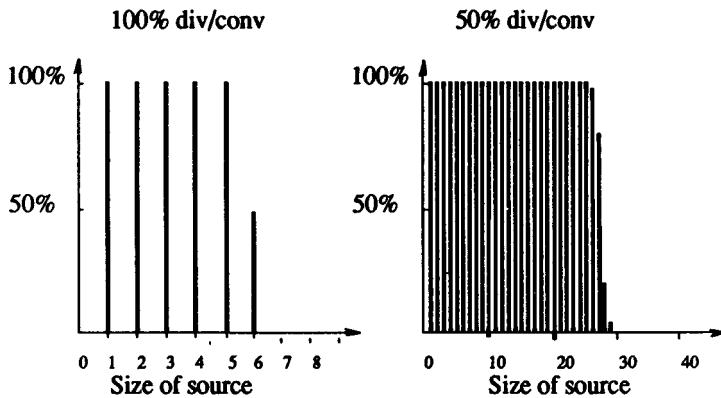


Figure 12.1. Probability of finding a diverging/converging link for a random network of 20000 neurons and a connection probability of 0.25.
 a) For a complete (100%) link.
 b) For a partial (50%) diverging/converging link (After M. Abeles).

grows to 25 neurons if instead of full contacts one is happy with a 50% connectivity. In the context of low activities of the network as a whole, the synchronous firing of a set of neurons will trigger the firing of the group of neurons it is closely linked to, thus starting the ignition of the whole chain of activities, the synfire chain. One notes the subtlety of the way information is processed by such a state dynamics. It is not certain however that this system is particularly robust. Delays several hundred milliseconds long imply chains of at least a hundred links. The existence of such long chains is still to be proved. There also exist other possible explanations of synchronous activities such as the automatic retrieval of temporal sequences of patterns.

12.4 Computing with attractors versus computing with flows of information

We have considered two main types of neuronal architectures, layered feedforward networks and fully connected networks. In the former, the information is processed layer by layer, the system having no activity of its own. Slight fluctuations in the input layer, owing for example to synaptic noise (the system being isolated from its environment), may give

rise to very large output fluctuations: in feedforward systems memory manifests itself only when inputs are present. Feedforward neural networks are essentially transducers, transforming inputs into convenient outputs, much as in accordance with the philosophy of behaviorism.

In fully connected networks on the other hand, the internal dynamics automatically brings the system towards a steady state, which materializes a state it has memorized in an earlier experience. The state of a fully connected network is therefore meaningful, even though the input units are inactive. This second architecture, with its possibilities of reflexive activities, is more appealing. It seems to open the way to autonomous, self-organizing systems.

Anatomical observations give indications of the existence of both architectures:

The sensory pathways are clearly organized in layers corresponding to different types of cells. Very schematically, the visual tract is made of a layer of photoreceptors, followed by a layer of bipolar cells. Then a layer of ganglion cells synapses on the various layers of the lateral geniculate nucleus. The tract proceeds towards the visual cortex where it divides along various paths corresponding to the parvo-cellular pathway, which seems to be involved in shape and color processing and to the magnocellular pathway which processes motion information. The signal is apparently analyzed in deeper and deeper structures. In reality the search seems bottomless and the neurophysiologist is disappointed at not being able to point out a definite structure where all this information would be gathered and processed, and where a final decision would be taken regarding at least the identification of the incoming signal.

There are about 10^8 photoreceptors in the retina and only 10^6 fibers running from the eyes to the cortex, a very large fan-in factor. The optic nerve is the input gate for about 10^9 neurons in the visual cortex, corresponding to an even larger fan-out factor. As a whole, the number of afferents in the cortex is extremely small with respect to the number of intra-cortical contacts. To quote V. Braintenberg, the cortex is an organ which essentially self-communicates and feedbacks must be the rule. Feedback contacts have in fact been observed everywhere in the nervous system, even in the retina, where the function of multiplexiform cells, whose activities seem to depend on that of cortex, has only just begun to be unraveled. For quite a while the importance of feedback fibers has been underestimated. The existence of retroactive circuits does give rise to an uncomfortable situation, since one must give up the hope of studying the functioning of the brain step by step, layer by layer. Probably, feedback contacts are essential, but contrary to what most memory models assume the respective roles of feedback and feedforward contacts are not symmetrical. One observes that the synapsing of

neurons in cortical layers of fibers transmitting information from sensory to motor areas differs from that where fibers are transmitting information in the opposite direction. The architecture of cortical neural networks seems to be both strongly connected and oriented.

The fundamental difference between layered and recurrent architectures is that strongly connected networks allow for the existence of persistent states. This can be essential if some computations in a part of the brain have to await the result of the computation carried out by some other part, for the computation to be further pursued.

Are persistent states observed in the brain? There are positive indications that long-lasting states do exist. For example, T. Miyashita and H. Chang observed persistent states by recording the activities of a set of cortical temporal neurons in trained monkeys responding to complicated visual stimuli. The patterns of electrical activities triggered by a specific image persisted well after the removal of the stimulus (see Fig. 12.2).

It is likely that both structures and both types of activities coexist in the brain. Braitenberg stresses the fact that in 1 mm^2 of cortex there are as many synapses originating from outside cells as from inside neurons. This is about the size of a cortical column.

12.5 The issue of low neuronal activities

Among the criticisms raised by the models of neural networks we have exposed in the text, one of most serious is the question of the low firing rates one observes in neural tissues. The models predict two types of neuronal activities: for low noise levels a neuron fires either close to its maximum rate, about 250 Hz, or close to zero frequency. If the noise level increases all neurons tend to fire at an average frequency of say 125 Hz. What is really observed is that two types of activities (maybe three, see below) coexist, but the frequencies are about 20 and 5 Hz respectively. Using sparse coding in the type of networks we have studied so far is not the answer to the problem since if, on average, the activity is low in sparse coding models, the activities of some neurons still remain high. Two suggestions have been made to cope with this sort of difficulty. The first idea is that the shape of the response function may drastically modify the individual average activities of neurons. The other proposal, which is more appealing, is that one has to take more seriously into account the very structure of the neuronal tissue, in particular the existence of excitatory pyramidal neurons whose activities may be controlled by those of inhibitory interneurons.

1) The first approach has been suggested by Amit and Hopfield. We assume that one pattern has been imprinted in the network according

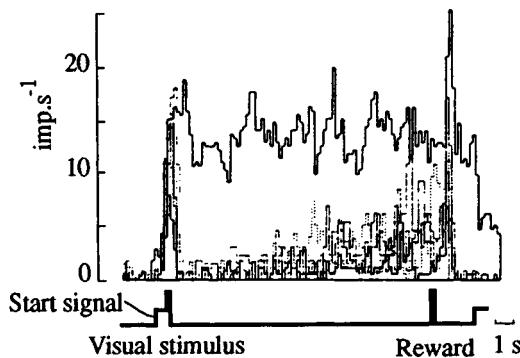
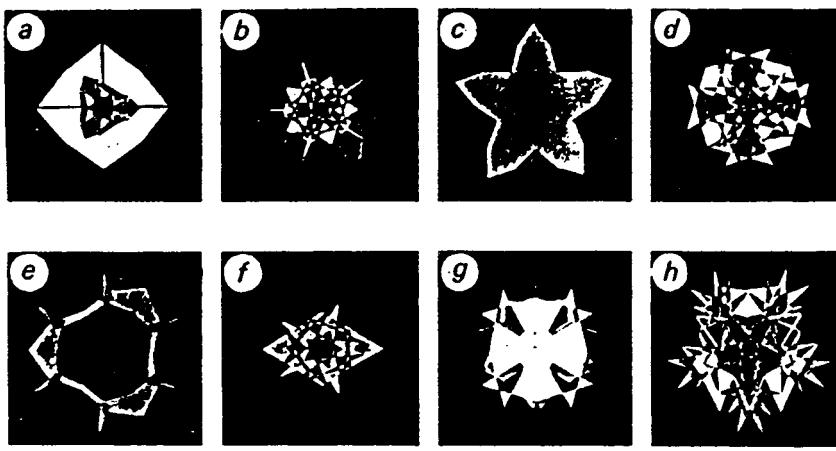


Figure 12.2. Response of a neuron in the anterior ventral temporal cortex in a visual task carried out by a macaque. The stimuli are fractal patterns (a to h). One observes that the neuron responds to all patterns, but that the activity is kept for long periods of time (here 16 seconds) only for very specific patterns (here pattern g). This is a strong indication that stimuli can trigger sustained activities (fixed points of activities) for a long while, an observation which is hard to account for by appealing to feedforward networks (After Miyashita and Chang).

to the usual symmetrical Hebbian rule:

$$J_{ij} = \frac{1}{N} \xi_i \xi_j.$$

The activities are given by $\sigma_i = \mathcal{S}\left(\frac{\beta}{N} \xi_i \sum_j \xi_j \sigma_j\right)$, where the response function $\mathcal{S}(x)$ is defined by

$$\mathcal{S}(x) = \begin{cases} 0 & \text{if } x < \theta, \\ \text{increasing to 1} & \text{if } x > \theta. \end{cases}$$

Distinguishing the *active sites* with $\xi_i = +1$ from the *passive sites* with $\xi_i = -1$, the activities obey the following mean-field dynamical equations:

$$\begin{aligned} \frac{d\sigma^+}{dt} &= -\frac{1}{\tau} [\sigma^+ - \mathcal{S}(\frac{1}{2}\beta(\sigma^+ - \sigma^-))], \\ \frac{d\sigma^-}{dt} &= -\frac{1}{\tau} [\sigma^- - \mathcal{S}(-\frac{1}{2}\beta(\sigma^+ - \sigma^-))]. \end{aligned}$$

The fixed points of the equations are given by:

$$\begin{aligned} X &= \mathcal{S}(\frac{1}{2}\beta X) - \mathcal{S}(-\frac{1}{2}\beta X), \\ Y &= \mathcal{S}(\frac{1}{2}\beta X) + \mathcal{S}(-\frac{1}{2}\beta X), \end{aligned}$$

with $X = \sigma^+ - \sigma^-$ and $Y = \sigma^+ + \sigma^-$. For low enough noise levels the system has a stable solution $X = Y \neq 0$, which yields

$$\sigma^+ \neq 0, \quad \sigma^- = 0.$$

The activity σ^- on passive sites is zero, while the activity σ^+ on active sites is of the order of the response for membrane potentials slightly larger than the threshold value θ . This activity may be well below the upper limit of $\sigma^+ = 1$. It remains to be seen how the memory storage capacities of the network are modified by choosing such a shape of response functions.

2) Other models have been put forward by Sompolinsky and Rubin on the one hand and by Amit and Treeves on the other. They rest on the effect of inhibitory neurons which tend to damp out the activities of pyramidal neurons. Here we present the model of Sompolinsky.

The network is made of two pools of interacting neurons. The first pool is comprised of N excitatory neurons $S_i \in \{0, 1\}$ whose interactions J_{ij} are modifiable on learning. The second pool is comprised of N inhibitory neurons $S_i^{\text{in}} \in \{0, 1\}$ whose negative interactions $-J' = -1$ are fixed. The interactions of excitatory on inhibitory neurons $J = +1$

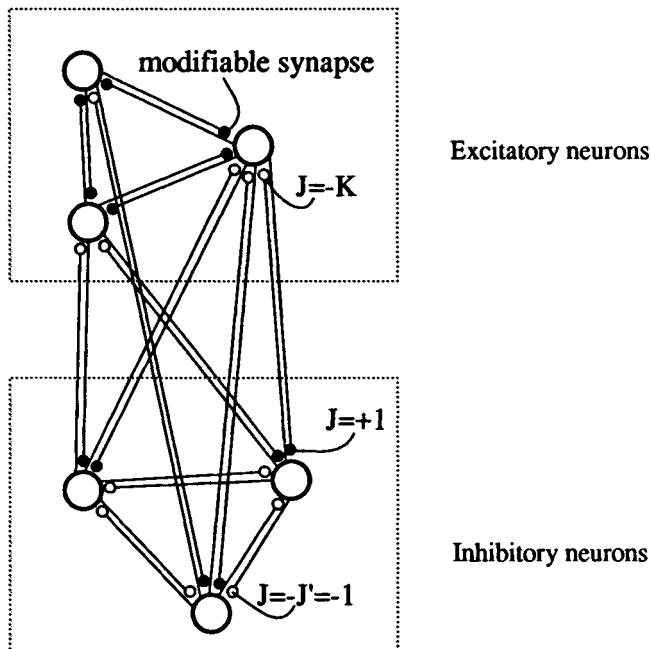


Figure 12.3. A simple model of cortical tissue which accounts for low firing rates of neurons (After Sompolinsky and Rubin).

and the interactions of inhibitory on excitatory neurons $-K$ are also fixed (Fig. 12.3). The strengths of modifiable synapses are assumed to follow the Willshaw model (section 7.5.5):

$$J_{ij} = \frac{1}{N} \mathbf{1} \left(\sum_{\mu=1}^P S_i^\mu S_j^\mu \right), \quad S_i^\mu \in \{0, 1\}.$$

As we know, for the Willshaw model to be interesting it is necessary to use sparse coding:

$$\sum_i S_i^\mu = f N, \quad \text{with } f \ll 1.$$

As usual, one considers the overlap between the running state and one of the memorized states μ . It is convenient to introduce three order parameters, namely average activities on active sites i^+ (those with $S_i^\mu = +1$), on passive sites i^- (with $S_i^\mu = -1$) and on inhibitory

neurons i^{in} :

$$\begin{aligned} S^+ &= \frac{1}{Nf} \sum_{i \in \text{exc}} S_i^\mu \langle S_i \rangle, \\ S^- &= \frac{1}{Nf(1-f)} \sum_{i \in \text{exc}} (1 - S_i^\mu) \langle S_i \rangle, \\ S^{\text{in}} &= \frac{1}{N} \sum_{i \in \text{in}} S_i. \end{aligned}$$

We do not want to give a detailed analysis of the model. Let us only mention a few results:

- The properties of the system are determined by the sign of the parameter $\theta - (K - 1)$.
- The case with $\theta < K - 1$ is the interesting one. Then the steady overlaps are given by

$$S^- \simeq 0, \quad S^+ = \frac{\theta}{K-1} < 1, \quad S^{\text{in}} = fS^+ \ll S^+.$$

- Depending on θ and K the activity S^+ may be very low. This means that only a small fraction S^+ of active sites i^+ of patterns S^μ are actually in state $S_i = +1$.

It is worth noting that any permutation of excitations on the active sites i^+ gives rise to the same S^+ and that it is easy for the network to wander among these degenerated states even at a low noise level. This phenomenon is related to the paradigm of frustration in spin glasses. The dynamics therefore creates continuously moving patterns of activities among those which are compatible with S^+ . On average the firing rate of a given neuron i^+ is low with bursting activities that appear from time to time, a fact that is not incompatible with experimental observations.

For the sake of concreteness we give here the equations which drive the dynamics of overlaps. According to Eqs (3.49) they are written as

$$\begin{aligned} \frac{dS^+}{dt} &= -\frac{1}{\tau} (S^+ - \mathcal{S}(\beta h^+)), \\ \frac{dS^-}{dt} &= -\frac{1}{\tau} (S^- - \mathcal{S}(\beta h^-)), \\ \frac{dS^{\text{in}}}{dt} &= -\frac{1}{\tau} (S^{\text{in}} - \mathcal{S}(\beta h^{\text{in}})), \end{aligned}$$

where the local fields are

$$\begin{aligned} h^+ &= S^+ + (1 - C)S^- - \frac{K}{f} S^{\text{in}} + \theta, \\ h^- &= (1 - C)(S^+ + S^-) - \frac{K}{f} S^{\text{in}} + \theta, \\ h^{\text{in}} &= S^+ + (1 - f)S^- - \frac{1}{f} S^{\text{in}}. \end{aligned}$$

θ is the threshold of excitatory neurons. Inhibitory neurons are not thresholded. C is the fraction of zero bonds:

$$C \simeq \exp -P f^2.$$

12.6 Learning and cortical plasticity

There is no direct experimental indication yet that synapses are plastic in the cortex itself. This is not because the cortical synapses are not modifiable on learning, but rather because it is extremely difficult to carry out experiments on plasticity in the cortex. Evidence for synaptic modifications have mainly been found in hippocampus on the one hand and in cerebellum on the other hand. Plasticity is of paramount importance to the alleged functions of these structures, long-term memory and motion coordination.

Among the learning rules discussed so far the back-propagation is certainly one of the less likely. How can we imagine that a neural network could implement the abstract operations involved in this algorithm? Let us consider the task of reaching an object. The visual input is defined in the frame of the retina, whereas the motion of the arm is determined with respect to the position of the body in space, that of the head in particular. An efficient motion therefore implies that a system automatically carries out the transformation of visual to head coordinates. The head coordinates are given by eye-position sensors.

Experiments carried out on trained monkeys by D. Zipser and R. Andersen show that an area of the posterior parietal cortex, called area 7a, contains three main types of neurons: the first type (about 15% of neurons of the area) responds to eye-position only, the second type (21% of neurons) responds to visual stimulations only and the third type (57%) to both sorts of inputs. This type codes the position of eyes with respect to that of the head.

The authors simulate this system by training a three layer feedforward network using the back-propagation algorithm. The inputs are made of two sets of neurons, an 8×8 array of retina cells and a 4×8 array of eye-position coding cells. The output is an array of 8×8 head-position coding cells. The hidden layer is made of 25 cells. After training the pattern of the activities of the hidden units and of the shapes of the receptive fields closely resembles that observed in experiments (see Fig. 12.4). These results are rather subjective and they do not imply that back-propagation is the way the cortex organizes its information. It suggests however that the brain tends to select optimal solutions through error correction. This makes it necessary for information to flow backwards from output to input units. Feedback fibers could be used for that purpose.

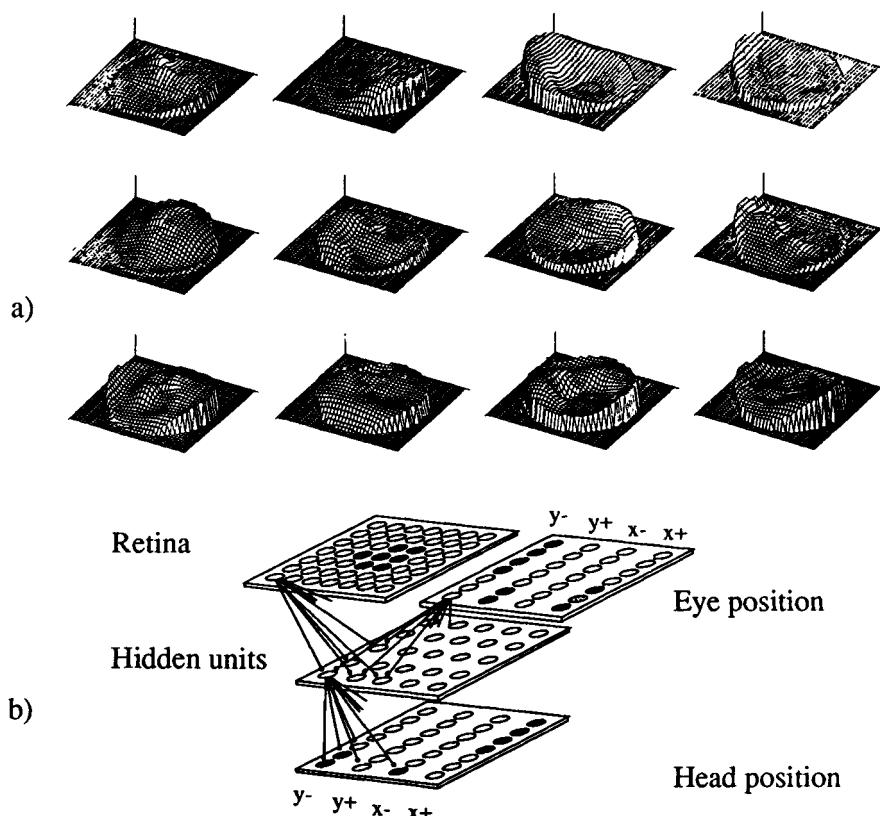


Figure 12.4 a) and b). Modelling the activity of the oculo-motor area 7a of a monkey (After Zipser and Andersen).
 a) The receptive fields actually measured in area 7a of a monkey.
 b) The three-layer feedforward model trained according to the back-propagation algorithm.

12.7 Taking the modular organization of the cortex into account

We have seen in Chapter 2 that the cortex is organized along a hierarchical architecture of neurons, microcolumns, columns, maps and areas. It is interesting to consider that such a structure aims at making the number of stored items as large as possible. Let us assume that a network of N neurons is made of an assembly of M modules of $n = N/M$ units. The capacity of one module is $p_c = \alpha n$.

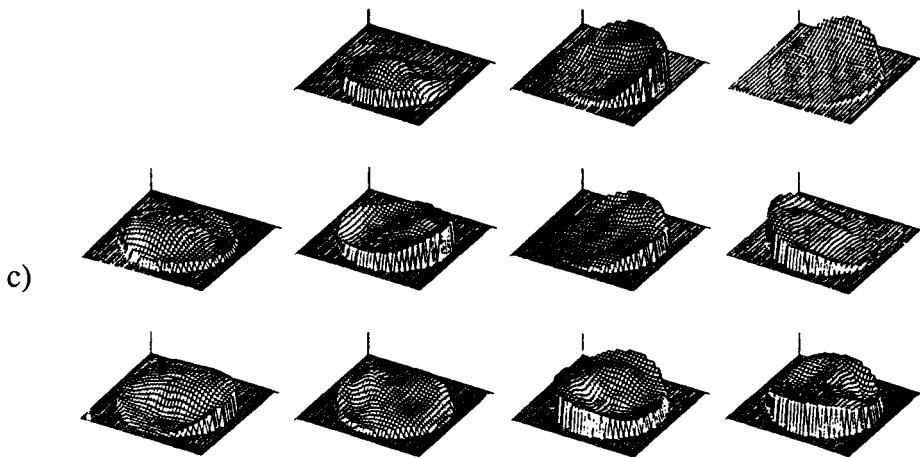


Figure 12.4.c. Modelling the activity of the occulo-motor activity in area 7a of a monkey (After Zipser and Andersen). The figure shows the receptive fields observed on the hidden units of the model.

If the modules are loosely connected the number of patterns the network is able to store is

$$P_c = (\alpha n)^M = (\alpha n)^{N/n}.$$

This number is maximum when

$$\frac{dP_c}{dn} = \frac{N}{n^2}(1 - \log(\alpha n)) = 0,$$

which gives

$$\alpha n = e \quad \text{or} \quad n = \frac{e}{\alpha}.$$

It is worth noting that the optimal size of the modules does not depend on the number N of neurons. With $n = 100$ the result leads to $\alpha = 0.027$, which is far from the maximum capacity allowed by the perceptron rule for example. However, we saw that constrained synaptic efficacies lower the capacity and the figure is not unlikely. On the other hand, the capacity of a module is $p_c = \alpha n = e \simeq 3$ patterns independent of its size. This naturally leads to models with modules as the basic units. The internal state of a unit can take three values, say $-1, 0, +1$. The simplest way of giving a meaning to these states is to assume that the module responds positively or negatively or is indifferent to the input it experiences.

These notions have been applied to a model of the training of Boolean networks by Aleksander (see section 8.3.3). They have been introduced

into neural networks by Shaw in the form of his *trion model*. A trion is a three-state module where the state 0 is strongly degenerate. The dynamics of the model is extremely complicated. A similar approach is advocated by Y. Burnod, who gives a more functional meaning to the state 0. For this author the modules are either ‘silent’ with an average activity of its constituent neurons close to 0 Hz, ‘active’ with an average activity of about 20 Hz, or ‘attentive’ with an activity of 5 Hz. An active module turns the states of modules it is connected to from silent to attentive. If inputs reinforce the attentive states the modules become active. According to this model the activity of the brain is mainly an exploration of possible responses through attentive modules, which are confirmed or invalidated by subsequent inputs. This is an attractive idea which is fairly well supported by electrophysiological studies (the existence of three states of activity for a module) and by psychological observations (the way children learn long words, for example). It needs to be more clearly formalized.

12.8 Higher-order processing: the problem of artificial intelligence

In their attempt to mimic human reasoning and behavior, workers in artificial intelligence naturally use high-level symbols as their basic units. Implicit in their approach is the assumption that these symbols are somewhere materialized in the central nervous system. The extreme view is that a symbol is associated with a neuron. This is the localist model whose semantic net is an example: each neuron is a grandmother cell. Obviously this view is untenable since this system is very sensitive to the destruction of a small amount of neurons, but the model seems to be salvaged if the neuron is replaced by the module as the basic unit of the net (see section 10.6.3).

Clearly, in biological systems the input signals are processed in many parallel pathways. Color, edge detection, motion and depth are treated by different sets of neurons, mainly in parallel. In monkeys more than 10 different visual maps have been observed. This can be considered as an efficient way of quickly processing information. However, once the features have been detected it seems likely that all this information will gather for the recognition process to be completed. For example, if color and shape are processed separately how is it possible for the color of an object to stick to its shape whatever the position of the object on the retina? There is experimental evidence collected by Julesz and Treisman that association of shapes with colors is carried out in series, but does this mean that the association process is to be precisely located?

Consider the way an image is transmitted to the brain. Each retina is

made of two parts, a left part and a right part. The boundary between the two parts is very acute, since it spreads in the retina over a range of less than half a millimeter. The left parts of both eyes, which receive information from the right half of the visual field, send their signals to the left cortical hemisphere of the brain and the two right parts, which receive information from the left half of the visual field, send their signals to the right hemisphere. Despite this complicated geometry and the fact that half of the visual field is treated by one hemisphere and the other by the other hemisphere, images are never interpreted as two half-images placed side by side. It is only when a hemisphere is impaired by a tumor or by a stroke that a half-image disappears and that the patient becomes aware of the existence of the vertical retinal boundaries. The corpus callosum is a sort of a gigantic bus made of about 8×10^8 fibers, which conveys a huge amount of signals between the two hemispheres. To avoid the spreading of epileptic seizures from one hemisphere to the other, a callostomy, that is a section of the corpus callosum, has sometimes been performed. In callostomized patients the two halves of images are clearly processed along completely separated pathways. Despite this, they still see images as unique entities. Only very special protocols are able to detect the impairment of corpus callosum. This shows that our notion of a symbol, as a reduction of a great number of instantiations to a unique, well-organized entity which contains them all, can in fact be flawed. A symbol could result from the activity of a large number of independent, competing, low-level systems which tend to express their states concurrently. If the result of such an anarchic behavior seems to be simple this is only because there are constraints on the output of the network which eliminate all the details of the activity of the various systems and it may well happen that nothing resembling a symbol can be found in the cortex either.

This philosophy is inherent in the models of learning, in layered networks in particular, which we have studied in the preceding sections. A problem can be stated in terms of symbols if the rules of the problem can be specified and the rules, in turn, come from the way a network is able to generalize. Generalization has been defined as the ability of a system which has been trained with a set of examples to respond properly to an example not belonging to the set. It is certain that the generalization abilities of a network will strongly depend on its architecture. Going back to symbolism, one can claim that the types of symbolic levels a system is able to achieve (if any) will strongly depend on its architectural constraints (that is, on the genetic material).

12.9 Concluding remarks

This text is open-ended. It has been written at a time of high activity in neuronal modeling. It is likely that many ideas that have been put forward here will be dismissed in the future, whereas some others have probably not been given the emphasis they might have deserved. More questions have been raised than answers given. All the same, if the field is the focus of such widespread interest this is due to the fact that, for the first time, it appears that elusive properties of central nervous systems, such as memory (memories), attention, reasoning or even motivation, could be modeled in hard networks. We are still very far from considering psychology as a chapter of physics, but a bridge has been launched between the two disciplines.

It is clear however that the most urgent problem in the field of neural networks is the lack of the usual and necessary dialectics between the theoretical speculations and the experimental evidences which could support or disprove them. It is also obvious that the status of this dialectics is more delicate here than it is in physics, for example. The ideal situation would be one where neurobiological observations would make clear which basic bricks are necessary to build an efficient theory of neural networks, and experimental psychology would bring clear-cut observations which the theory has to account for. At this stage the theory could even be endowed with predictive power. However, the present situation is not that satisfactory.

The main thrust in neurobiology is towards molecular biology and biochemistry. Many researchers are engaged in deciphering the sequences of amino acids which make up the neuroreceptor proteins. Others are working on the complicated intertwined loops of chemical reactions which are involved in the functioning of the neuronal cells, in the generation of electro-physiological signals, in synaptic transmission and in all the regulations which are necessary for the system to work. No doubt this research is of paramount importance. Nevertheless there is other research, often of a more classical nature, which seems to be less popular and which would bring a host of relevant information to the theory of neural networks.

Let us consider anatomy, for example. A great deal of anatomical information exists about the architecture of neural networks, but the available data very often favour the exceptional over the general, whereas the modeler only needs to know about the general properties of the neuronal tissues. These are not always properly outlined.

- The role of the various types of neurons is an example. There is no doubt that the pyramidal neurons are the most important cells of the cortical tissue: but how numerous are they? Figures ranging from 60

to 85% of the overall neuronal population can be found in the literature. Can we consider all the other types of neurons as mere interneurons, the role of which is to provide an effective interaction between the pyramidal neurons? Or do they have some specific role, modifying the global properties of the pyramidal neurons and so transforming a network into a different one? For example, could it be possible for a diffuse network to exist which would control attention as in the searchlight hypothesis?

- The study of the anatomy of synapses, that is of their modifications after training, is clearly of major importance for the modeling of learning. Of particular interest would be the study of those modifications, if any, on short time scales some seconds or even shorter. Are the synaptic efficacies really plastic in the cortex? If this is so, as seems probable, what is the mechanism responsible for the modifications? Is it pre- or post-synaptic? On what time scale? Is NMDA (N-methyl-D-aspartate) the key protein in long-term memorization processes? How long does the effect last? Does a crystallization effect exist? Is plasticity limited to excitatory synapses? Are the synaptic efficacies limited by cellular or by synaptic resources?

- An issue of great significance is the determination of the role of the various parts of the CNS. What exactly is the cerebellum made for? The LGN's, the hippocampus, ...? Which parts of the cortex are involved in a given task, in which order, to which amount? How localized or delocalized are items memorized in the cortex? Does something more or less resembling a semantic network exist? Is it hierarchically organized? Are other parameters, such as the thresholds, also modifiable, either on learning or by some global feedback mechanisms?

If neurobiology is the source of our knowledge of the structures of neural networks and the basic mechanisms which make up their dynamics, experimental psychology is the supplier of countless series of experiments which the theory must strive to explain. Closer contacts between theoreticians and psychologists are necessary to determine which sort of experiments could bring an answer to a well-posed problem. Let us cite for example the experiment which showed that humans are less apt at solving tasks which imply an XOR logic than tasks which imply AND logic. The idea of this experiment has been suggested by the failure of the perceptron to solve the XOR problem.

Finally, it must be emphasized that there is a need to reinforce the connections between artificial intelligence and the neural networks approach. The links have proved to be very loose up to now. Systematic applications of neural networks to the problems tackled by AI must be undertaken.

The theory of neural networks we have developed in this text is very far from being satisfactory. The problem of the role of delays in

the neuronal dynamics has not been treated properly. Among other important issues, one can cite the training of hidden units, the relation between learning and topology that is the organization of the information in the network, the role of the columnar organization of the cortex, the relation between learning and delays, the place of bursting or oscillatory activities The great number of problems to be solved leaves much scope for the fertile imagination of researchers from many disciplines.

REFERENCES

BOOKS

- Abeles, M. — *Local Cortical Circuits: An electrophysiological study.* — Studies in Brain Function, vol. 6, Springer, Berlin (1982).
- Ackley, D.H. — *A Connectionist Machine for Genetic Hill Climbing.* — Kluwer, London (1987).
- Albert, A. — *Regression and the More Penrose Pseudo-Inverse.* — Academic Press, N.Y. (1972).
- Alberts, A., Bray, D., Lewis, J., Raff, M., Roberts, K., Watson, J.D. — *Molecular Biology of the Cell.* — Garland, N.Y. (1983).
- Albus, J. — *Brains, Behavior and Robotics.* — BYTE Books, McGraw Hill, Peterborough, N.H. (1981).
- Aleksander, I. — *Neural Computing Architectures.* — Kogan Page, London (1988).
- Alkon, D.L. — *Memory Traces in the Brain.* — Cambridge University Press, N.Y. (1987).
- Amit, D. — *Modelling Brain Function.* — Cambridge University Press, Cambridge (1989).
- Anderson, J.R. — *Language, Memory, and Thought.* — Erlbaum, Hillsdale, N.J. (1976).
- Anderson, J.R. — *The Architecture of Cognition.* — Harvard University Press, Cambridge, MA (1983).
- Arbib, M.A. — *The Metaphorical Brain.* — Wiley, N.Y. (1972 and 1989).
- Arbib, M.A. — *Brains, Machines, and Mathematics.* — Springer, Berlin (1987).
- Arbib, M.A., Amari, S. — *Dynamic Interactions in Neural Networks. — Models and Data,* Springer, Berlin (1989).
- Atlan, H. — *Theories of Immune Networks.* — Springer, Berlin (1989).
- Balian, R. — *Du Microscopique au Macroscopique.* — vol. 1, Presses de l'École Polytechnique, Ellipses, Paris (1982).
- Bareiss, R. — *Examplar-Based Knowledge Acquisition.* — Academic Press, N.Y. (1990).
- Barr, A., Cohen, P.R., Feigenbaum, E.A. — *The Handbook of Artificial Intelligence.* — vols 1, 2 and 3, Addison-Wesley, Wokingham (1986).
- Basar, E., Bullock, T.H. — *Brain Dynamics.* — Springer, Berlin (1989).
- Beer, R. — *Intelligence as Adaptive Behavior.* — Academic Press, London (1990).
- Beltrami, E. — *Mathematics for Dynamic Modeling.* — Academic Press, London (1987).

- Benzécri, J.P. — *L'Analyse des Données*. — vol. 1: *La Taxinomie*, Dunod, Paris (1984).
- Binder, K. — *Monte-Carlo Methods in 'Statistical Physics'*. — Topics in Current Physics, vol. 7, Springer, Berlin (1979).
- Birkmayer, W. — *Understanding the Neurotransmitters: Key to the Workings of the Brain*. — Springer, Berlin (1989).
- Bolk, L. — *Natural Language Parsing Systems*. — Springer, Berlin (1987).
- Bolk, L. — *Computational Models of Learning*. — Springer, Berlin (1987).
- Bourret, P., Reggia, J., Samuelides, M. — *Les Réseaux Neuronaux*. — Technea, Paris (1990).
- Bower, G.H., Hilgard, E.J. — *Theories of Learning*. — Prentice-Hall, N.Y. (1981).
- Bradzil, P.B., Konolige, K. — *Machine Learning, Meta-Reasoning and Logics*. — Kluwer, London (1989).
- Braitenberg, V. — *On the Texture of Brains: An Introduction to Neuroanatomy for the Cybernetically Minded*. — Springer, Berlin (1977).
- Brink, J.R., Haden, C.R. — *The Computer and the Brain*. — North Holland, Amsterdam (1989).
- Bunge, M. — *The Mind Body Problem: A Psychobiological Approach*. — Pergamon Press, Oxford (1980).
- Burnod, Y. — *An Adaptive Network: The Cerebral Cortex*. — Collection Biologie Théorique, vol. 3, Masson, Paris (1989).
- Buser, P., Imbert, M. — *Vision*. — Hermann, Paris (1987).
- Callatay, A.M. — *Natural and Artificial Intelligence*. — Elsevier, N.Y. (1986).
- Carey, G.F. — *Parallel Supercomputing*. — Wiley, N.Y. (1989).
- Cattell, R.B. — *Intelligence: Its Structure, Growth and Action*. — Elsevier, N.Y. (1987).
- Chandler, D. — *Introduction to Modern Statistical Mechanics*. — Oxford University Press, Oxford (1987).
- Changeux, J.P. — *L'Homme Neuronal*. — Fayard, 'Le Temps des Sciences', Paris (1983). — *Neuronal Man*. — Pantheon Books, N.Y. (1985).
- Churchland, P.M. — *Matter and Consciousness: A Contemporary Introduction to the Philosophy of Mind*. — MIT Press, Cambridge, MA (1988).
- Chvatal, V. — *Linear Programming*. — Freeman, N.Y. (1983).
- Cruz, C.A. — *Understanding Neural Networks*. — Elsevier, N.Y. (1988).
- Delacour, J. — *Conditionnement et Biologie*. — Masson, Paris (1981).
- Delgado-Frias, J.G., Moore, W.R. — *VLSI for Artificial Intelligence*. — Kluwer, London (1989).
- Dickinson, A. — *Contemporary Animal Learning Theory*. — Cambridge University Press, Cambridge (1980).
- Droz, R., Richelle, M. — *Manuel de Psychologie: Introduction à la psychologie scientifique*. — Mardaga, Bruxelles (1985).
- Duda, R., Hart, P. — *Pattern Classification and Scene Analysis*. — Wiley, N.Y. (1973).

- Durbin, R., Miall, C., Mitchinson, G. — *The Computational Neuron*. — Addison-Wesley, Wokingham (1989).
- Eccles, J.C. — *The Physiology of Synapses*. — Springer, Berlin (1964).
- Eccles, J.C., Ito, M., Szentagothai, J. — *The Cerebellum as a Neuronal Machine*. — Springer, Berlin (1987).
- Edelman, G. — *Neural Darwinism*. — Basic Books, N.Y. (1987).
- Edelman, G., Mountcastle, V.B. — *The Mindful Brain: Cortical Organization and the Group Selective Theory of Higher Brain Functions*. — MIT Press, Cambridge, MA (1978).
- Edelman, G., Gall, W.E., Cowan, W.M. — *New Insights into Synaptic Function*. — Wiley, N.Y. (1985).
- Erickson, G., Ray Smith, C. — *Maximum Entropy and Bayesian Methods in 'Sciences and Engineering'*. — vol. 1: Foundations, vol. 2: Applications, Kluwer, London (1988).
- Faurre, P. — *Cours d'Analyse Numérique: Notes d'Optimisation*. — École Polytechnique, Paris (1984).
- Feller, W. — *Introduction to Probability Theory and its Applications*. — vol. I, Wiley, N.Y. (1968).
- Fodor, J.A. — *The Language of Thought*. — Crowell, N.Y. (1975).
- Fodor, J.A. — *The Modularity of Mind*. — MIT Press, Cambridge, MA (1983).
- Gantmacher, F. — *Applications of the Theory of Matrices*. — Wiley, N.Y. (1959).
- Garey, M.R., Johnson, D.S. — *Computers and Intractability: A Guide to the Theory of NP-Completeness*. — Freeman, San Francisco, CA (1979).
- Gazzaniga, M.S. — *Perspectives in 'Memory Research'*. — MIT Press, Cambridge, MA (1988).
- Geszti, T. — *Physical Models of Neural Networks*. — World Scientific, Singapore (1990).
- Goles, E., Martinez, S. — *Neural and Automata Networks: Dynamical Behavior and Applications*. — Kluwer, London (1990).
- Gondran, M., Minoux, M. — *Graphes et Algorithmes*. — 2^{ème} edn., Eyrolles, Paris (1985).
- Gordon, G.H. — *The Psychology of Learning and Motivation*. — Academic Press, N.Y. (1973).
- Gordon, I.E. — *Theories of Visual Perception*. — Wiley, N.Y. (1988).
- Grossberg, S. — *Studies of Brain and Mind*. — Kluwer, London (1982).
- Grossberg, S. — *The Adaptive Brain*. — vols. 1 and 2, Elsevier, N.Y. (1984).
- Grossberg, S. — *Neural Networks and Natural Intelligence*. — MIT Press, Cambridge, MA (1988).
- Haken, H., Stadler, M. — *Synergetics of Cognition*. — Springer, Berlin (1990).
- Hall, J.F. — *The Psychology of Learning*. — Lippincott, Philadelphia (1966).
- Harth, E. — *Windows on the Mind: Reflections on the Physical basis of Consciousness*. — William Marrow, N.Y. (1982).

- Hebb, D.O. — *The Organization of Behavior: A Neuropsychological Theory.* — Wiley, N.Y. (1949).
- Hebb, D.O., Donderi, D.C. — *Textbook of Psychology.* — Lawrence Erlbaum, Hillsdale, N.J. (1987).
- Hecht-Nielsen, R. — *Neurocomputing: Non-Algorithmic Information.* — Addison-Wesley, Wokingham (1990).
- Heim, R., Palm, G. — *Theoretical Approaches to Complex Systems.* — Springer, Berlin, N.Y. (1978).
- Hertz, J. — *Introduction to the Theory of Neural Computation. Santa-Fe Lecture Notes Series in 'Computer and Neural Systems'.* — Addison-Wesley, London (1980).
- Hillis, W.D. — *The Connection Machine.* — MIT Press (1985)., — *La Machine à Connexions.* — Masson, Paris (1989).
- Hockney, R., Jesshope, C. — *Parallel Computers.* — Adam Hilger, Bristol (1981).
- Holden, A. — *Models of the Stochastic Activity of Neurones.* — Lecture Notes in Biomathematics, vol. 12, Springer, Berlin (1976).
- Hoppensteadt, F. — *An Introduction to the Mathematics of Neurons.* — Cambridge University Press, Cambridge (1986).
- Hubel, D. — *Eye, Brain, and Vision.* — Scientific American Library, N.Y. (1988).
- Isaacson, D., Madsen R. — *Markov Chains, Theory and Applications.* — Robert Krieger, Malabar, FA. (1976).
- Ito, M. — *The Cerebellum and Neural Control.* — Raven, N.Y. (1984).
- Jacob, F. — *The Possible and the Actual.* — University of Washington Press, Seattle (1982).
- Jardine, N., Sibson, R. — *Mathematical Taxonomy.* — Wiley, N.Y. (1971).
- Johnson, R.C., Brown, C. — *Cognizers.* — Wiley, N.Y. (1988).
- Kahnna, T. — *Foundations of Neural Networks.* — Addison-Wesley, N.Y. (1989).
- Kamp, Y., Hasler, M. — *Réseaux de Neurones Récurrsifs pour Mémoires Associatives.* — Presses Polytechniques Romandes, Lausanne (1990).
- Kandel, E. — *Cellular Basis of Behavior: An introduction to behavioral neurobiology.* — Freeman, San Francisco, CA (1976).
- Katz, B. — *Nerve, Muscle and Synapse.* — McGraw-Hill, N.Y. (1966).
- Klimasauskas, C.C. — *The 1989 Neurocomputing Bibliography.* — MIT Press, Cambridge, MA (1989).
- Klivington, K. — *The Science of Mind.* — MIT Press, Cambridge, MA (1989).
- Klopff, H. — *The Hedonistic Neuron: A Theory of Memory, Learning, and Intelligence.* — Hemisphere, Washington, D.C. (1982).
- Koch, C., Segev, I. — *Methods in Neuronal Modeling.* — MIT Press, Cambridge, MA (1989).
- Kohonen, T. — *Associative Memory. A System-Theoretical Approach.* — Communication and cybernetics, vol. 17, Springer, N.Y. (1978).

- Kohonen, T. — *Content-Addressable Memories*. — Springer, Berlin (1980).
- Kohonen, T. — *Self-Organization and Associative Memory*. — Springer, N.Y. (1984 and 1989).
- Kowalski, R. — *Logic for Problem Solving*. — Artificial Intelligence Series, Ed. Nilsson, N., Elsevier, N.Y. (1979).
- Kullback, S. — *Information Theory and Statistics*. — Wiley, N.Y. (1959).
- Levy, W., Anderson, J.A., Lehmkuhle, S. — *Synaptic Modification, Neuron Selectivity and Nervous System Organization*. — Lawrence Erlbaum, London (1985).
- MacCulloch, W.S. — *Embodiments of Mind*. — MIT Press, Cambridge, MA (1965 and 1988).
- MacGregor, R. — *Neural and Brain Modeling*. — Academic Press, San Diego (1987).
- Mackintosh, N.J. — *The Psychology of Animal Learning*. — Academic Press, N.Y. (1974).
- Marr, D. — *Vision*. — Freeman, San Francisco, CA (1982).
- Marler, P., Terrace, H. — *The Biology of Learning*. — Springer, Berlin (1984).
- Maynard Smith, J. — *Evolution and the Theory of Games*. — Cambridge University Press, Cambridge (1982).
- Mead, C., Conway, L. — *Introduction to VLSI Systems*. — Addison-Wesley, London (1980).
- Mead, C. — *Analog VLSI Implementation of Neural Systems*. — Kluwer, London (1989).
- Mead, C. — *Analog VLSI and Neural Systems*. — Addison-Wesley, N.Y. (1989).
- Mel, B. — *Connectionist Robot Motion Planning*. — Academic Press, London (1990).
- Mesarovic, M., Macko, D., Takahara, Y. — *Theory of Hierarchical Multilevel Systems*. — Academic Press, N.Y. (1970).
- Mézard, M., Parisi, G., Virasoro, M. — *Spin Glass Theory and Beyond*. — Lecture Notes in Physics, vol. 9, World Scientific, Singapore (1987).
- Mill, P.J. — *Comparative Neurobiology*. — Edward Arnold, London (1982).
- Minsky, M.L., Papert, S.A. — *Perceptrons: An Introduction to Computational Geometry*. — MIT Press, Cambridge, MA (1969 and 1988).
- Morse, P.M., Feschbach, D. — *Methods of Mathematical Physics*. — vol. I, McGraw Hill, N.Y. (1953).
- Naim, P., Davalo, E. — *Des Réseaux de Neurones*. — Eyrolles, Paris (1989).
- Nelson, P. — *Logique des Neurones et du Système Nerveux: Essai d'analyse théorique des données expérimentales*. — Maloine-Doin, Paris (1978).
- Nelson, R.J. — *The Logic of Mind*. — Kluwer, London (1989).
- O'Keefe, J., Nadel, L. — *The Hippocampus as a Cognitive Map*. — Clarendon Press, Oxford (1978).
- O'Shea, T., Eisenstadt, M. — *Artificial Intelligence*. — Harper & Row, N.Y. (1984).

- Palm, G. — *Neural Assemblies: An Alternative Approach to Artificial Intelligence, Studies of brain function.* — Springer, N.Y. (1982).
- Papadimitriou, C.H., Streightz, K. — *Combinatorial Optimization Algorithms and Computing.* — Prentice-Hall, Englewood Cliffs, N.Y. (1982).
- Partee, C., Hartson, C., Maren, A., Pap, R. — *Handbook of Neural Computing Applications.* — Academic Press, N.Y. (1990).
- Pavlov, J.P. — *Conditioned Reflexes.* — Dover, N.Y. (1960).
- Pfeifer, R. — *Connectionism in Perspective.* — North Holland, Amsterdam (1989).
- Piaget, J. — *The Origins of Intelligence in Children.* — International University Press, N.Y. (1952).
- Piaget, J. — *The Construction of Reality in the Child.* — Basic Books, N.Y. (1954).
- Pieroni, G.G. — *Issues on Machine Vision.* — Springer, Berlin (1989).
- Pinker, S. — *Learnability and Cognition.* — MIT Press, Cambridge, MA (1989).
- Pinker, S., Mehler, J. — *Connections and Symbols.* — MIT Press, Cambridge, MA (1988).
- Popper, K.R., Eccles, J.C. — *The Self and Its Brain.* — Springer, Berlin (1985).
- Posner, M.I. — *The Foundations of Cognitive Science.* — MIT Press, Cambridge, MA (1990).
- Ramon y Cajal, S. — *Histologie du Système Nerveux.* — vols I et II, Maloine, Paris (1911), CSIC, Madrid (1972).
- Reichardt, W.E., Poggio, T. — *Theoretical Approaches in Neurobiology.* — MIT Press, Cambridge, MA (1981).
- Reuchlin, M. — *Psychologie.* — 6^{ème} ed., PUF, Fondamental, Paris (1986).
- Richards, B., Berthke, I., Oberlander, J., Van des Does, J. — *Temporal Representation of Inference.* — Academic Press, N.Y. (1989).
- Robert, F. — *Discrete Iterations: A Metric Study.* — Springer, Berlin (1980).
- Rosenblatt, F. — *Principles of Neurodynamics: Perceptron and the Theory of Brain Mechanisms.* — Spartan, Washington, D.C. (1961).
- Rumelhart, D.E., MacClelland, J. and the PDP Research Group. — *Parallel Distributed Processing: Explorations in the microstructure of cognition.* — vol. 1: *Foundations*, vol. 2: *Psychological and Biological Models*, MIT Press, Cambridge, MA (1988).
- Sakarovitch, M. — *Optimisation Combinatoire.* — vol. 1: *Graphes et Programmation Linéaire*, vol. 2: *Programmation Discrète*, Hermann, Paris (1984).
- Schmitt, F., Worden, F., Adelman, G., Dennis, S. — *The Organization of the Cerebral Cortex.* — MIT Press, Cambridge, MA (1981).
- Schuster, H. — *Deterministic Chaos.* — Physik-Verlag, Weinheim RFA (1984).
- Serra, R., Zanarini, G. — *Complex Systems and Cognitive Processes.* — Springer, Berlin (1989).
- Shank, R., Colby, M. — *Computer Models of Thought and Language.* — Freeman, San Francisco, CA (1973).

- Soucek, B. and M. — *Neural and Massively Parallel Computers: The Sixth Generation.* — Wiley, N.Y. (1988).
- Soucek, B. — *Neural and Concurrent Real-Time Systems.* — Wiley, N.Y. (1989).
- Szentagothai, J., Arbib, M. — *Conceptual Models of Neural Organization.* — MIT Press, Cambridge MA (1975).
- Thomas, R. — *Kinetic Logic: A Boolean Approach to the Analysis of Complex Regulatory Systems.* — Lecture Notes in Biomathematics, vol. 29, Springer, Berlin (1979).
- Tikhonov, A.N., Arsenin, V.Y. — *Solutions of Ill-Posed Problems.* — Winston, Washington, D.C. (1977).
- Treleaven, P.C. — *Parallel Computers.* — Wiley, N.Y. (1989).
- Vitushkin, A.G. — *Theory of Transmission and Processing of Information.* — Pergamon Press, N.Y. (1961).
- Von Neumann, J. — *The Computer and the Brain.* — Yale University Press, Westford, MA (1958).
- Wanatabe, S. — *Pattern Recognition: Human and Mechanical.* — Wiley, N.Y. (1985).
- Wechsler, H. — *Computational Vision.* — Academic Press, London (1990).
- Weisbuch, G. — *Dynamique des Systèmes Complexes: Une Introduction aux Réseaux d'Automates.* — Savoirs actuels, Éditions du CNRS, Paris (1989).
- Winograd, J., Flores, F. — *L'Intelligence Artificielle en Question.* — PUF, Paris.
- Winston, P.H. — *Artificial Intelligence.* — Addison-Wesley, London (1979 and 1984).
- Wolfram, S. — *Theory and Applications of Cellular Automata.* — World Scientific, Singapore (1986).
- Woody, C.D. — *Memory, Learning and Higher Functions: A Cellular View.* — Springer, N.Y. (1982).
- Zornetzer, S.F., Davis, J.L., Lau, C. — *Introduction to Neural and Electronic Networks.* — Academic Press, N.Y. (1990).

PROCEEDINGS

- Almeida, L.B. — *EURASIP Workshop on Neural Networks, Sesimbra, Portugal.* — Lecture Notes in Computer Science, vol. 412, Springer, Berlin (1990).
- Amari, S., Arbib, M.A. — *Competition and Cooperation in Neural Nets.* — Lecture Notes in Biomathematics, vol. 45, Springer, Berlin (1982).
- Anderson, D.Z. — *Proceedings of the IEEE Conference on Neural Information and Processing Systems.* — IEEE, Denver (November 1987).
- Anderson, J.A., Rosenfeld, E. — *Neurocomputing.* — MIT Press, Cambridge, MA (1987).
- Arbib, M.A., Hanson, A.R. — *Vision, Brain and Cooperative Computation.* — MIT Press, Cambridge, MA (1987).

- Balian, R., Maynard, R., Toulouse, G. — *Ill-Condensed Matter*. — Les Houches Summer School, Session 1978, North Holland, Amsterdam (1979).
- Basar, E., Flor, H., Haken, H., Mandell, A.J. — *Synergetics of the Brain*. — Proc. Int. Symp. Synergetics, Springer, N.Y. (1978).
- Bienenstock, E., Fogelman Soulie, F., Weisbuch, G. — *Disordered Systems and Biological Organization*. — NATO series F: Computer and Systems Sciences, vol. 20. Springer, N.Y. (1986).
- Byrne, J., Berry, W. — *Neural Models of Plasticity. Experimental and Theoretical Approaches*. — Academic Press, London (1989).
- Caianiello, E.R. — *Physics of Cognitive Processes*. — World Scientific, Singapore (1987).
- Casti, J.L., Kariqvist, A. — *Real Brains, Artificial Minds*. — Elsevier, N.Y. (1987).
- Cotterill, R. — *Computer Simulation in Brain Science*. — Copenhagen Aug. 20–22 (1986). Cambridge University Press, Cambridge (1987).
- Cowan, J.P., Sharp, D.H. — *Neural Nets*. — Los Alamos National Laboratory, Los Alamos, NM (1987).
- Ciba. — *Functions of the Septo-Hippocampal System*. — Elsevier, N.Y. (1978).
- DARPA. — *Neural Networks Study*. — Armed Forces Communications and Electronics Association (AFCEA), International Press, Fairfax, VA. (1988).
- Delacour, J. — *Neurobiologie des Comportements*. — Hermann, Paris (1984).
- Delacour, P. — *Apprentissage et Mémoire*. — Masson, Paris (1987).
- Demongeot, J., Goles, E., Tchuente, M. — *Dynamical Systems and Cellular Automata*. — Academic Press, London (1985).
- Denker, J.S. — *Proceedings of the AIP Conference No. 151. — Neural Networks for Computing*, American Institute of Physics (1986).
- Eckmiller, R., Von der Malsburg, C. — *Neural Computers*. — NATO ASI, series F, vol. 41, Springer, Berlin (1989).
- Edelman, G., Gall, E., Cowan, M. — *Synaptic Function*. — Wiley, N.Y. (1987).
- Fayad, R., Rodriguez-Vargas, A., Violini, G. — *Proceedings of the First Latin American School on Biophysics*. — World Scientific, Singapore (1987).
- Freeman, H. — *Machine Vision for Inspection and Measurement*. — Academic Press, N.Y. (1989).
- Finke, R.A. — *Principles of Mental Imagery*. — MIT Press, Cambridge, MA (1989).
- Gardner, E. — *Special Issue in Memory of Elizabeth Gardner (1957–1988)*. — Journal of Physics A, Mathematical and General, 22, vol. 12 (1989).
- Haken, H. — *Neural and Synergetic Computers*. — A symposium held at Schloss Elmau (Bavaria), Springer, Berlin (1988).
- Hawkins, R., Bower, G. — *Computational Models of Learning in Simple Neural Systems*. — Academic Press, London (1989).
- Hinton, G.E., Anderson, J.A. — *Parallel Models of Associative Memory*. — Lawrence Erlbaum, Hillsdale, NJ (1981).
- IEEE Transactions on Systems, Man and Cybernetics*. — Special Issue on Neural and Sensory Information Processing, SMC-13 (September–October 1983).

- IEEE Proceedings. — *First International Conference on Neural Networks.* — IEEE, San Diego, CA (1987);
- Second International Conference on Neural Networks.* — IEEE, San Diego, CA (1988).
- Lawler, E.L. — *The Traveling Salesman Problem.* — Wiley, N.Y. (1984).
- Martinez, J., Kesner, R. — *Learning and Memory: A Biological View.* — Academic Press, London, (1986).
- Michalski, R., Carbonnel, J., Mitchell, T. — *Machine Learning: An Artificial Intelligence Approach.* — Springer, Berlin, (1984).
- Nadel, L., Culicover, P., Cooper, L.A., Harnish, R.M. — *Neural Connections, Mental computation.* — MIT Press, Cambridge, MA (1989).
- Neyrinck, J. — *Journées d'Electronique: Réseaux de Neurones Artificiels.* — Presses Polytechniques Romandes, Lausanne, Switzerland (1989).
- Nicolini, C. — *Modeling and Analysis in Biomedecine.* — World Scientific, Singapore (1984).
- Olshen, R., Breiman, L., Friedman, J., Stone, C. — *Classification and Regression Trees.* — Wadsworth International Group, Belmont, CA (1984).
- Parten, C., Harston, C., Moren, A., Pap, R. — *Handbook of Neural Computing Applications.* — Academic Press, London (1990).
- Personnaz, L., Dreyfus, G. — *Neural Networks from Models to Applications.* — IDSET, Paris (1989).
- Pinsker, H., Willis, W. — *Information Processing in the Nervous System.* — Raven Press, N.Y. (1980).
- Rosenzweig, M., Bennett, E. — *Neural Mechanisms of Learning and Memory.* — MIT Press, Cambridge, MA (1976).
- Schmitt, F., Worden, F., Adelman, G., Dennis, S. — *The Organization of the Cerebral Cortex.* — Proceedings of a neuroscience research colloquium, MIT Press, Cambridge, MA (1981).
- Shwab, E., Nusbaum, H. — *Pattern Recognition by Humans and Machines.* — vol. I: *Speech perception*, vol. II: *Visual Perception*, Academic Press, London (1986).
- Spillmann, L., Werner, J. — *Visual Perception: The Neurophysiological Foundations.* — Academic Press, London (1990).
- Theumann, W., Köberle, R. — *Neural Networks and Spin Glasses.* — World Scientific, Singapore (1990).
- Uhr, L. — *Pattern Recognition.* — Wiley, N.Y. (1966).
- Van Hemmen, L., Morgenstern, I. — *Glassy Dynamics and Optimization.* — Proceedings of the 1986 Heidelberg Colloquium, Springer Lecture Notes in Physics (1987).
- Wise, S.P. — *Higher Brain Functions: Recent Explorations of the Brain's Emergent Properties.* — Wiley-Interscience, N.Y. (1987).

ARTICLES

- Abbott, L.F., Arian, Y. — *Storage capacity of generalized networks.* — Phys. Rev. A 36, 5091–5094 (1987).
- Abbott, L.F. — *Learning in neural network memories.* — Preprint, Brandeis University, Waltham, MA, N° BRX-TH-274 (1989).
- Abbott, L.F., Kepler, T.B. — *Optimal learning in neural networks memories.* — Preprint, Brandeis University, Waltham, MA, N° BRX-TH-255 (1989).
- Abbott, L.F. — *Universality in the space of interactions for network models.* — Preprint, Brandeis University, Waltham, MA, N° BRX-TH-263 (1989).
- Abbott, L.F. — *Modulation of function and gated learning in a network memory.* — Preprint, Brandeis University, Waltham MA, N° BRX-TH-284 (1990).
- Abeles, M., Goldstein, M.H. — *Multiple spike train analysis.* — Proc. IEEE 65, 762–773 (1977).
- Abeles, M. — *Neural codes for higher brain functions.* — In ‘Information Processing by the Brain’, Markowitsch, H.J. Ed., Hans Huber, Toronto (1990).
- Abu-Mostafa Y. — *Information theory, complexity, and neural networks.* — IEEE Commns Mag. 25–28 (November 1989).
- Abu-Mostafa, Y., Psaltis, D. — *Optical neural computers.* — Sci. Am. 256, 88–95 (March 1987).
- Abu-Mostafa, Y., St-Jacques, J.M. — *Information capacity of the Hopfield model.* — IEEE Trans. IT 31, 461–464 (1985).
- Ackley, D., Hinton, T., Sejnowski, T. — *A learning algorithm for Boltzman machines.* — Cog. Sci. 9, 147–169 (1985).
- Albus, J.S. — *A theory of the cerebellar function.* — Math. Biosci. 10, 25–61 (1971).
- Albus, J.S. — *Mechanisms of planning and problem solving in the brain.* — Math. Biosci. 45, 247–293 (1979).
- Aleksander, I. — *Mind, brain, structure and function.* — Kybernetes, Thales, London, 11, 249–253 (1982).
- Alkon, D.L. — *Voltage-dependent calcium and potassium ion conductances: A contingency mechanism for an associative learning model.* — Science 205, 810–816 (1979).
- Alkon, D., Lederhendler, I., Shoukimas, J. — *Primary changes of membrane currents during retention of associative learning.* — Science 215, 693–695 (1982).
- Alspector, J. — *Neural-style microsystems that learn.* — IEEE Commns. Mag. 29–36 (November 1989).
- Alspector, J., Allen, R., Hu, V., Satyanarayana, S. — *Stochastic learning networks and their electronic implementation.* — In Proc. IEEE Conf. ‘Neural Information Processing Systems’, Anderson, D.Z. Ed., Denver, pp. 9–21 (1987).
- Altman, J. — *Images in and of the brain.* — Nature 324, 405 (1986).

- Amari, S. — *Characteristics of randomly connected threshold element networks and network systems.* — IEEE Trans. 59, 35–47 (1971).
- Amari, S. — *A method of statistical neurodynamics.* — Kybernetik 14, 201–215 (1974).
- Amari, S. — *Neural theory of association and concept-formation.* — Biol. Cybern. 26, 175–185 (1977).
- Amari, S.I., Yoshida, K., Kanatani, K.I. — *A mathematical foundation for statistical neurodynamics.* — SIAM J. Appl. Math. 33, 95–126 (1977).
- Amari, S.I. — *Learning patterns and pattern sequences by self-organizing net of threshold elements.* — IEEE Trans. C-21, (1972).
- Amari, S.I., Lumsden, C.J. — *A mathematical approach to semantic network development.* — Bull. Math. Biol. 47, 629–650 (1985).
- Amit, D.J., Gutfreund, H., Sompolinsky, H. — *Spin-glass models of neural networks.* — Phys. Rev. A 32, 1007–1018 (1985).
- Amit, D.J., Gutfreund, H., Sompolinsky, H. — *Storing infinite numbers of patterns in a spin-glass model of neural networks.* — Phys. Rev. Lett. 55, 1530–1533 (1985).
- Amit, D., Gutfreund, H., Sompolinsky, H. — *Information storage in neural networks with low levels of activity.* — Phys. Rev. A 35, 2293–2303 (1987).
- Amit, D., Gutfreund, H., Sompolinsky, H. — *Statistical mechanics of neural networks near saturation.* — Ann. Phys. 173, 30–67 (1987).
- Amit, D. — *Neural networks counting chimes.* — Proc. Natl Acad. Sci. USA 85, 2141–2145 (1988).
- Anderson, A., Palca, J. — *Who knows how the brain works?* — Nature 335, 489–491 (1988).
- Anderson, A. — *Learning from a computer cat.* — Nature 331, 657–659 (1988).
- Anderson, C.H., van Essen, D.C. — *Shifter circuits: A computational strategy for dynamic aspects of visual processing.* — Proc. Natl Acad. Sci. USA 84, 6297–6301 (1987).
- Anderson, J.A. — *Two models for memory organization using interacting traces.* — Math. Biosci. 8, 137–160 (1970).
- Anderson, J.A. — *A theory of recognition of items from short memorized lists.* — Psychol. Rev. 80, 417–438 (1973).
- Anderson, J.A. — *Cognitive and psychological computation with neural models.* — IEEE Trans. SMC-13, 799–815 (1983).
- Anderson, J.A. — *Cognitive and psychological computation with neural models.* — IEEE Trans. SMC-13, 799–815 (1983).
- Anderson, P.W. — *Lectures on amorphous systems.* — In 'Ill Condensed Matter', Les Houches Session XXXI (1978), Balian, R. Ed., North Holland, Amsterdam, pp. 161–261 (1979).
- Anninos, P., Beck, B., Csermely, T., Harth, E., Pertile, G. — *Dynamics of neural structures.* — J. Theor. Biol. 26, 121–148 (1970).
- Anninos, P.A. — *Cyclic modes in artificial neural nets.* — Kybernetik 11, 5–14 (1972).

- Anninos, P., Kokkinidis, M., Skouras, A. — *Noisy neural nets exhibiting memory domains*. — J. Theor. Biol. 109, 581–594 (1984).
- Anninos, P., Kokkinidis, M. — *A neural net model for multiple memory domains*. — J. Theor. Biol. 109, 95–110 (1984).
- Anninos, P., Argyrakis, P., Skouras, A. — *A computer model for learning processes and the role of the cerebral commissures*. — Biol. Cybern. 50, 329–336 (1984).
- Ans, B., Herault, J., Jutten, C. — *Adaptive neuromimetic architectures: Detection of primitives*. — In 'Cognitiva 85', Cesta-AFCET Eds., Paris, pp. 593–597 (1985).
- Anshelevich, V., Amirkian, B., Lukashin, A., Frank-Kamenetskii, M. — *On the ability of neural networks to perform generalization by induction*. — Preprint, to appear in Biol. Cybern. (1989).
- Arbib, M.A., Kilmer, W.L., Spinelli, R.N. — *Neural models and memory*. — In 'Neural Mechanisms and Memory', Rosenzweig, M., Benett, E. Eds., MIT Press, Cambridge, MA (1976).
- Arbib, M., Amari, S.I. — *Sensori-motor transformations in the brain (with a critique of the tensor theory of cerebellum)*. — J. Theor. Biol. 112, 123–155 (1985).
- Arbib, M. — *Artificial intelligence and brain theory: Unities and diversities*. — Ann. Biomed. Eng. 3, 238–274 (1975).
- Bahren, J., Gulati, S., Zak, M. — *Neural learning of constrained nonlinear transformations*. — IEEE Trans. C-22, 67–76 (1989).
- Baldi, P., Venkatesh, S. — *Number of stable points for spin glasses and neural networks of higher orders*. — Phys. Rev. Lett. 58, 913–916 (1987).
- Balian, R., Vénéroni, M., Balazs, N. — *Relevant entropy versus measurement entropy*. — Europhys. Lett. 1, 1–5 (1986).
- Ballard, D., Hinton, G., Sejnowski, T. — *Parallel visual computation*. — Nature 306, 21–26 (1983).
- Baranyi, A., Feher, O. — *Intracellular studies on cortical synaptic plasticity*. — Exp. Brain Res. 41, 124–134 (1981).
- Baranyi, A., Feher, O. — *Synaptic facilitation requires paired activation of convergent pathways in the neocortex*. — Nature 290, 413–415 (1981).
- Barlow, H.B. — *Single units and sensation: A neuron doctrine for perceptual psychology*. — Perception 1, 371–394 (1972).
- Barnes, D. — *Neural models yield data on learning*. — Science 236, 1628–1629 (1988).
- Barriomuevo, G., Brown, T. — *Associative long-term potentiation in hippocampal slices*. — Proc. Natl Acad. Sci. USA 80, 7347–7351 (1983).
- Barto, A., Sutton, R., Brouwer, P. — *Associative search network: A reinforcement learning associative memory*. — Biol. Cybern. 40, 201–211 (1981).
- Barto, A., Sutton, R. — *Simulation of anticipatory responses in classical conditioning by a neuron-like adaptive element*. — Behav. Brain Sci. 4, 221–235 (1982).

- Barto, A.G., Sutton, R.S., Anderson, C.W. — *Neuronlike elements that can solve difficult learning control problems.* — IEEE Trans. SMC-13, 834-846 (1983).
- Baum, E. — *Intractable computations without local minima.* — Phys. Rev. Lett. 57, 2764-2767 (1986).
- Bear, M., Cooper, L., Ebner, F. — *A physiological basis for a theory of synapse modification.* — Science 237, 42-48 (1987).
- Beck, J. — *Textural segmentation, second order statistics, and textural elements.* — Biol. Cybern. 48, 125-130 (1983).
- Bernier, L., Castellucci, V., Kandel, E., Schwartz, J. — *cAMP determination in sensory neurons.* — J. Neurosci. 2, 1682-1691 (1982).
- Bialek, W., Zee, A. — *Statistical mechanics and invariant perception.* — Phys. Rev. Lett. 58, 741-744 (1987).
- Bialek, W., Zee, A. — *Understanding the efficiency of human perception.* — Phys. Rev. Lett. 61, 1512-1515 (1988).
- Bienenstock, E. — *Cooperation and competition in central neurons system development: A unifying approach.* — In 'Synergetics of the brain', Proc. of the international symposium on synergetics. Basar, E., Flohr, H., Haken, H., Mandell, A.J., Springer Eds., N.Y. (1978).
- Bienenstock, E., Von der Malsburg, C. — *A neural network for invariant pattern recognition.* — Europhys. Lett. 4, 121-126 (1987).
- Bienenstock, E., Cooper, L., Munro, P. — *Theory for the development of neuron selectivity: Orientation specificity and binocular interaction in visual cortex.* — J. Neurosci. 2, 32-48 (1982).
- Bienenstock, E., Doursat, R. — *Elastic matching and pattern recognition in neural networks.* — In 'Neural Networks: From Models to Applications', Personnaz, L., Dreyfus, G. Eds., IDSET, Paris (1989).
- Blozovski, D. — *L'hippocampe et le comportement.* — La Recherche, Paris 175, 330-336 (1986).
- Bohn, G. — *A structure for associative information processing.* — Biol. Cybern. 29, 193-200 (1978).
- Bonomi, E., Lutton, J.L. — *Le recuit simulé.* — Pour la Science, Paris 129, 68-77 (1988).
- Borisyuk, G., Borisyuk, R., Kirilov, A., Kovalenko, E., Kryukov, V. — *A new statistical method for identifying interconnections between neuronal network elements.* — Biol. Cybern. 52, 301-306 (1985).
- Bounds, D.G. — *A statistical mechanical study of Boltzmann machines.* — J. Phys. A: Math. Gen. 20, 2133-2145 (1987).
- Bourret, P., Gaspin, C., Samuelides, M. — *Affectation dynamique des ressources d'un satellite en opération par une machine de Boltzmann à sensibilisation.* — Preprint, ONERA-CERT, Toulouse (1988).
- Braddick, O. — *Vision in humans and computers.* — Nature 323, 201 (1986).
- Brady, R.M. — *Optimization strategies gleaned from biological evolution.* — Nature 317, 804-806 (1985).

- Brannan, J., Boyce, W. — *Spatially localized interactive neural populations I: A mathematical model.* — Bull. Math. Biol. 43, 427–446 (1981).
- Bray, A.J., Moore, M.A. — *Some observations on the mean-field theory of spin glasses.* — J. Phys. C: Solid St. Phys. 13, 419–434 (1980).
- Bray, A.J., Moore, M.A., Young, A.P. — *Weighted averages of TAP solutions and Parisi's $q(x)$.* — J. Phys. C: Solid St. Phys. 17, L155–L160 (1984).
- Braitenberg, V. — *Cortical architectonics: General and areal.* — In 'Architectonics of the Cerebral Cortex', Brazier, M.A., Petsche, H. Eds., p. 443, Raven Press, N.Y. (1978).
- Braitenberg, V. — *Cell assemblies in the cerebral cortex.* — In 'Theoretical approaches to complex systems', Heim, R., Palm, G. Eds, p. 171, Springer, Berlin, N.Y. (1978).
- Brennan, S. — *Caricature generator: The dynamic exaggeration of faces by computer.* — Leonardo 18, 170–178 (1985).
- Brown, T. — *Neural networks for switching.* — IEEE Commns Mag. 72–81 (November 1989).
- Browne, J.C. — *Parallel architectures for computer systems.* — Phys. Tod. 2835 (May 1984).
- Bruce, A., Gardner, E., Wallace, D. — *Static and dynamic properties of the Hopfield model.* — J. Phys. A: Math. Gen. 20, 2909–2934 (1987).
- Buhmann, J., Divko, R., Schulten, K. — *Associative memory with high information content.* — Preprint, Technische Universität, München (1988).
- Buhmann, J., Schulten, K. — *Associative recognition and storage in a model network of physiological neurons.* — Biol. Cybern. 54, 319–335 (1986).
- Buhmann, J., Schulten, K. — *Influence of noise on the function of a 'physiological' neural network.* — Biol. Cybern. 56, 313–327 (1987).
- Buhmann, J., Schulten, K. — *Noise-driven temporal association in neural networks.* — Europhys. Lett. 4, 1205–1209 (1987).
- Buhmann, J., Lange, J., Von der Malsburg, C., Vorbrüggen, J., Würtz, R. — *Object recognition in the dynamic link architecture: Parallel implementation on a transputer network.* — In 'Neural Networks: A Dynamical Systems Approach to Machine Intelligence', Kosko, B. Ed., Prentice-Hall, Englewood Cliffs, NJ (1990).
- Buisseret, P., Singer, W. — *Proprioceptive signals from extraocular muscles gate experience-dependent modifications of receptive fields in the kitten visual cortex.* — Exp. Brain Res. 51, 443–450 (1983).
- Bullock, T. — *Comparative neuroscience holds promise for quiet revolution.* — Science 225, 473–477 (1984).
- Burnod, Y., Korn, H. — *Consequences of stochastic release of neurotransmitters for network computation in the central nervous system.* — Proc. Natl Acad. Sci. USA 86; 352–356 (1989).
- Caianiello, E.R. — *Outline of a theory of thought processes and thinking machines.* — J. Theor. Biol. 1, 204–235 (1961).

- Caianiello, E.R., de Luca, A., Ricciardi, L.M. — *Reverberations and control of neural networks.* — Kybernetik 4, 10–18 (1967).
- Canning, A., Gardner, E. — *Partially connected models of neural networks.* — Preprint, Edinburgh 88/433 (cf. J. Phys. A (1988)).
- Carew, T.J., Castellucci, V.F., Kandel, R.R. — *An analysis of dishabituation and sensitization of the gill-withdrawal reflex in Aplysia.* — Int. J. Neurosci. 2, 79–98 (1971).
- Carew, T., Walters, E., Kandel, E. — *Classical conditioning in a simple withdrawal reflex in Aplysia californica.* — J. Neurosci. 1, 1426–1437 (1981).
- Carew, T., Walters, E., Kandel, E. — *Classical conditioning in a simple withdrawal reflex in Aplysia californica.* — J. Neurosci. 1, 1426–1437 (1981).
- Carnevali, P., Patarnello, S. — *Exhaustive Thermodynamical Analysis of Boolean Learning Networks.* — Europhys. Lett. 4, 1199–1204 (1987).
- Carpenter, G. — *Neural network models for pattern recognition and associative memory.* — Review article, Neural Networks 2, 243–257 (1989).
- Casdagli, M. — *Nonlinear prediction of chaotic time series.* — Physica 35 D, 335–356 (1989).
- Case, J., Fisher, P. — *Long-term memory modules.* — Bull. Math. Biol. 46, 295–326 (1984).
- Casinius, J., Van Hemmen, J.L. — *A polynomial time algorithm in general quadratic programming and ground-state properties of spin glasses.* — Europhys. Lett. 1, 319–326 (1986).
- Cerny, V. — *Multiprocessor system as a statistical ensemble: A way towards general purpose parallel processing and MIMD computers.* — Preprint, Inst. of Phys. and Biophys., Comenius Univ., Bratislava (1983).
- See also a preprint entitled: *Annealing algorithm and parallel processing: An option to approach pattern recognition problems* (1984).
- Chalfie, M., Sulston, J., White, J., Southgate, E., Thomson, N., Brenner, S. — *The neural circuit for touch sensitivity in Caenorhabditis elegans.* — J. Neurosci. 5, 956–964 (1985).
- Changeux, J.P., Heidmann, T. — *Allosteric receptors and molecular models of learning.* — In 'New Insights into Synaptic Function', Edelman, G., Gall, W.E., Cowan, W.M. Eds., Wiley, N.Y. (1985).
- Changeux, J.P., Heidmann, T., Piette, P. — *Learning by selection.* — In 'The Biology of Learning', Marler, P., Terrace, H. Eds., pp. 115–133, Springer, Berlin (1984).
- Changeux, J.P., Devillers-Thiéry, A., Chemouilli, P. — *Acetylcholine receptor: An allosteric protein.* — Science 225, 1335–1345 (1984).
- Changeux, J.P., Courrège, P., Danchin, A. — *A theory of the epigenesis of neural networks by selective stabilization of synapses.* — Proc. Natl Acad. Sci. USA 70, 1974–2978 (1973).
- Changeux, J.P., Danchin, A. — *Selective stabilization of developing synapses as a mechanism for the specification of neural networks.* — Nature 264, 705–712 (1976).

- Chay, T. — *Abnormal discharges and chaos in a neuronal model system.* — Biol. Cybern. 50, 301–311 (1984).
- Chen, K. — *A simple learning algorithm for the traveling salesman problem.* — Preprint, Brookhaven National Laboratory, Brookhaven, N.Y. (1989).
- Choi, M., Huberman, B. — *Dynamic behavior of non-linear networks.* — Phys. Rev. A 28, 1204–1206 (1983).
- Choi, M., Huberman, B. — *Collective excitations and retarded interactions.* — Phys. Rev. B 31, 2862–2866 (1985).
- Choi, M.Y., Huberman, B.A. — *Digital dynamics and the simulation of magnetic systems.* — Phys. Rev. B 28, 2547–2554 (1983).
- Chua, L.O., Yang, L. — *Cellular neural networks: Theory.* — IEEE Trans. CS-35, 1257–1272 (1988). — *II Applications.* — IEEE Trans. CS-35, 1273–1290 (1988).
- Churchland, P., Sejnowski, T. — *Perspectives on cognitive neuroscience.* — Science 242, 741–745 (1988).
- Clark, J. — *Statistical mechanics of neural networks.* — Phys. Rep. 158, 91–157 (1988).
- Clark, J., Rafelski, J., Winston, J. — *Brain without mind: Computer simulation of neural networks with modifiable neuronal interactions.* — Phys. Rep. 123, 215–273 (1985).
- Clark, J., Winston, J., Rafelski, J. — *Self-organization of neural networks.* — Phys. Lett. 102A, 207–211 (1984).
- Colby, K.M. — *Mind models: An overview of current work.* — Math. Biosci. 39, 159–185 (1978).
- Colding-Jorgensen, M. — *A model for the firing pattern of a paced nerve cell.* — J. Theor. Biol. 101, 541–568 (1983).
- Conrad, M. — *Microscopic-macroscopic interface in biological information processing.* — Biosyst. 16, 345–363 (1984).
- Cooley, A., Gielen, C. — *Delays in neural networks.* — Europhys. Lett. 7, 281–285 (1988).
- Cooper, L., Liberman, F., Oja, E. — *A theory for the acquisition and loss of neuron specificity in visual cortex.* — Biol. Cybern. 33, 9–28 (1979).
- Cooper, L. — *A possible organization of animal memory and learning.* — In Proc. Nobel Symp. Collective Prop. Phys. Syst., Lundquist, B., Lundquist, S. Eds. (London) 24, 252–264 (1973).
- Cortes C., Hertz, J.A. — *A network system for image segmentation.* — Preprint, Nordita, Copenhagen, № 89/5 S (1989).
- Cosnard, M., Goles, E. — *Dynamique d'un automate à mémoire modélisant le fonctionnement d'un neurone.* — C. R. Acad. Sci. Paris 299, 459–461 (1984).
- Cottrell, M., Fort, J.C. — *About the retinotopy: A self organizing process.* — Biol. Cybern. 53, 405–411 (1986).
- Cottrell, M., Fort, J.C. — *Etude d'un processus d'auto-organisation.* — Ann. Inst. Henri Poincaré 23, 1–20 (1987).
- Cover, T.M. — *Geometrical and statistical properties of systems of linear inequalities with applications to pattern recognition.* — IEEE Trans. EC-14, 326–334 (1965).

- Cowan, J.D. — *A statistical mechanics of nervous activity.* — In 'Lectures on Mathematics in the Life Sciences 2', Gerstenhaber, M. Ed., Am. Math. Soc. Providence, RI (1970).
- Cragg, B.G., Temperley, H.N.V. — *The organization of neurons: A cooperative analogy.* — *Electroencephalog. Clin. Neurophys.* 6, 85 (1954).
- Cragg, B.G., Temperley, H.N.V. — *Memory: The analogy with ferromagnetic hysteresis.* — *Brain* 78, 304 (1955).
- Crick, F. — *The recent excitement about neural networks.* — *Nature* 337, 129–132 (1989).
- Crick, F., Mitchinson, G. — *The function of dream sleep.* — *Nature* 304, 111–114 (1983).
- Crick, F. — *Function of the thalamic reticular complex: The searchlight hypothesis.* — *Proc. Natl Acad. Sci. USA* 81, 4586–4590 (1984).
- Crisanti, A., Amit, D.J., Gutfreund, H. — *Saturation level of the Hopfield model for neural networks.* — *Europhys. Lett.* 2, 337–341 (1986).
- Csermely, T.J., Harth, E., Lewis, N.S. — *The netlet theory and cooperative phenomena in neural networks.* — *J. Dyn. Syst. Meas. Control, ASME, Rochester, N.Y.*, 315–320 (September 1973).
- Dale, N., Schacher, S., Kandel, E.R. — *Long-term facilitation in Aplysia involves increase in transmitter release.* — *Science* 239, 282–285 (1988).
- Dammasch, I. — *Morphogenesis and properties of neuronal model networks.* — In 'Cybernetics and Systems '86', Trappi, R. Ed., Reidel, pp. 327–334 (1986).
- Dayoff, J. — *Distinguished words in data sequences: Analysis and applications to neural coding and other fields.* — *Bull. Math. Biol.* 46, 529–543 (1984).
- Dehaene, S., Changeux, J.P., Nadal, J.P. — *Neural networks that learn temporal sequences by selection.* — *Proc. Natl Acad. Sci. USA* 84, 2727–2731 (1987).
- Dehaene, S., Changeux, J.P. — *A simple model of prefrontal cortex function in delayed-response tasks.* — Preprint, to appear in *J. Cognitive Neurosci.* (1989).
- De Kwaadsteniet, J. — *Statistical analysis and stochastic modeling of neuronal spike-train activity.* — *Math. Biosci.* 60, 17–71 (1982).
- Delacour, J. — *Two neuronal systems are involved in a classical conditioning in the rat.* — *Neurosci.* 13, 705–715 (1984).
- Del Castillo, J., Katz, B. — *Quantal components of the end-plate potential.* — *J. Physiol.* 24, 560–573 (1954).
- Del Castillo, J., Katz, B. — *Statistical factors involved in neuromuscular facilitation and depression.* — *J. Physiol. (London)*, 124, 574–585 (1954).
- Del Guidice, P., Franz, S., Virasoro, M.A. — *Perceptron beyond the limit of capacity.* — *J. Phys., Paris* 50, 121–134 (1989).
- Demetrius, L. — *Statistical mechanics and population biology.* — *J. Stat. Phys.* 30, 709–753 (1983).
- Denker, J. — *Neural network models of learning and adaptation.* — *Physica* 22D, 216–232 (1986).

- Denker, J., Schwartz, D., Wittner, B., Solla, S., Hopfield, J.J., Howard, R., Jackel, L. — *Large automatic learning, Rule extraction, and generalization.* — Complex Syst. 1, 877 (1987).
- Deprit, E. — *Implementing recurrent back-propagation on the connection machine.* — Neural Networks 2, 295–314 (1989).
- Derrida, B., Flyvberg, H. — *Multivalley structure in Kauffman's model: Analogy with spin glasses.* — J. Phys. A: Math. Gen. 19, L1003–L1008 (1986).
- Derrida, B., Gardner, E., Zippelius, A. — *An exactly soluble asymmetric neural network model.* — Europhys. Lett. 4, 167–173 (1987).
- Derrida, B., Nadal, J.P. — *Learning and forgetting on asymmetric diluted neural networks.* — J. Stat. Phys. 49, 993–1009 (1987).
- Derrida, B., Weisbuch, G. — *Evolution of overlaps between configurations in random Boolean networks.* — J. Phys., Paris 47, 1297–1303 (1986).
- Derrida, B., Pomeau, Y. — *Random networks of automata: A simple annealed approximation.* — Europhys. Lett. 1, 45–49 (1986).
- De Werra, D., Hertz, A. — *Tabu search techniques.* — Oper. Res. Spektrum 11, 131–141 (1989).
- D'Humières, D., Huberman, B.A. — *Dynamics of self-organization in complex adaptive networks.* — J. Stat. Phys. 34, 361–379 (1984).
- Dickinson, A., Mackintosh, N.J. — *Classical conditioning in animals.* — Ann. Rev. Psychol. 29, 487–612 (1978).
- Diedrich, S., Opper, M. — *Learning of correlated patterns in spin-glass networks by local learning rules.* — Phys. Rev. Lett. 58, 949–952 (1987).
- Dodd, N. — *Graph matching by stochastic optimisation applied to the implementation of multilayer perceptrons on transputer networks.* — Parallel Computing (North Holland) 10, 135–142 (1989).
- Domany, E., Meir, R., Kinzel, W. — *Storing and retrieving information in a layered spin system.* — Europhys. Lett. 2, 175–185 (1986).
- Domany, E., Kinzel, W. — *Equivalence of cellular automata to Ising models and directed percolation.* — Phys. Rev. Lett. 53, 311–314 (1984).
- Domany, E. — *Neural networks: A biased overview.* — J. Stat. Phys. 51, 743–775 (1988).
- Dorizzi, B., Grammaticos, B., Le Berre, M., Pomeau, Y., Ressayre, E., Tallet, A. — *Statistics and dimension of chaos in differential delay systems.* — Phys. Rev. A 35, 328–339 (1987).
- Dotsenko, V.S. — *Fractal dynamics of spin glasses.* — J. Phys. C 18, 6023–6031 (1985).
- Dreyfus, G. — *Chaos et C.A.O. ou la méthode du 'recuit simulé'.* — AFCET/Interfaces, Paris 53, 4–9 (1987).
- Dunin-Barkowski, W., Larionova, N. — *Computer simulation of a cerebellar cortex compartment: I. General Principles and properties of a neural net.* — Biol. Cybern. 51, 399–406 (1985).
- Duranton, M., Sirat, J.A. — *Réseau de neurones: Bientôt un composant VLSI.* — Minis and Micros 323, 41–47 (Mai 1989).

Durbin, R., Willshaw, D.J. — *An analogue approach to the travelling salesman problem using an elastic net method.* — Nature 326, 689–691 (1987).

Easton, P., Gordon, P. — *Stabilization of Hebbian neural nets by inhibitory learning.* — Biol. Cybern. 51, 1–9 (1984).

Eccles, J.C. — *The modular operation of the cerebral neocortex considered as the material basis of mental events.* — Neurosci. 6, 1839–1856 (1981).

Edelman, G., Reeke, G. — *Selective networks capable of representative transformations, limited generalizations, and associative memory.* — Proc. Natl Acad. Sci. USA 79, 2091–2095 (1982).

Eigen, M. — *Self-organization of matter and the evolution of biological macromolecules.* — Naturwiss. 58, 465–523 (1971).

Erdi, P., Barna, G. — *Self-organizing mechanism for the formation of ordered neural mappings.* — Biol. Cybern. 51, 93–101 (1984).

Faith, D. — *Patterns of sensitivity of association measures in numerical taxonomy.* — Math. Biosci. 69, 199–207 (1984).

Feigel'man, M.V., Ioffe, L.B. — *Hierarchical nature of spin glasses.* — J. Physique Lett. 45, L475–L481 (1984).

Feigel'man, M.V., Ioffe, L.B. — *The augmented models of associative memory: Asymmetric interaction and hierarchy of patterns.* — Int. J. Mod. Phys. B1, 51–68 (1987).

Feldman, J.A. — *Dynamic connections in neural networks.* — Biol. Cybern. 46, 27–39 (1982).

Feldman, J.L., Cowan, J.D. — *Large-scale activity in neural nets. I: Theory with application to motoneuron pool responses.* — Biol. Cybern. 17, 29–38 (1975).

Feldman, J.L., Cowan, J.D. — *Large-scale activity in neural nets. II: A model for the brainstem respiratory oscillator.* — Biol. Cybern. 17, 39–51 (1975).

Feldman, J.L. — *Dynamic connections in neural networks.* — Biol. Cybern. 46, 27–39 (1982).

Fogelman-Soulié, F., Goles-Chacc, E., Weisbuch, G. — *Specific roles of the different Boolean mappings in random networks.* — Bull. Math. Biol. 44, 715–730 (1982).

Fogelman-Soulié, F., Gallinari, P., Le Cun, Y., Thiria, S. — *Automata networks and artificial intelligence.* — In ‘Automata Networks in Computer Science, Theory and Applications’, Fogelman-Soulié, F., Robert, Y., Tchuente, M. Eds., Manchester University Press (1987).

Fontanary, J., Meir, R. — *Mapping correlated gaussian patterns in a perceptron.* — J. Phys. A: Math. Gen. 22, L803–L808 (1989).

Forrest, B.M. — *Content-addressability and learning in neural networks.* — J. Phys. A: Math. Gen. 21, 245–255 (1987).

Forrest, B., Roweth, D., Stroud, N., Wallace, D., Wilson, G. — *Neural network models.* — Parall. Comp. 8, 71–83 (1988).

- Foster, D., Kahn, J. — *Internal representations and operations in the visual comparison of transformed patterns: Effects of pattern point-inversion, positional symmetry, and separation.* — Biol. Cybern. 51, 305–312 (1985).
- Fox, J.L. — *The brain's dynamic way of keeping in touch.* — Science 225, 820–821 (1984).
- Fox, G., Otto, S. — *Algorithms for concurrent processors.* — Phys. Tod., 50–59 (May 1984).
- Frégnac, Y., Imbert, M. — *Early development of visual cortical cells in normal and dark-reared kittens. Relationship between orientation selectivity and ocular dominance.* — J. Physiol. (London) 278, 27–44 (1978).
- Frégnac, Y. — *Cellular mechanisms of epigenesis in cat visual cortex.* — In 'Imprinting and Cortical Plasticity', Rauschecker, J.P. Ed. Wiley, N.Y. (1986).
- Frey, P.W., Sears, R. — *Model of conditioning incorporating the Rescorla-Wagner associative axiom, a dynamic attention process, and a catastrophe rule.* — Psychol. Rev. 85, 321–340 (1978).
- Frobenius, S.B. — Pressus Akad. Wiss. 471 (1908).
- Fukushima, K. — *A model of associative memory in the brain.* — Kybernetik 12, 58–63 (1973).
- Fukushima, K. — *Cognitron: A self-organizing multi-layered neural network.* — Biol. Cybern. 20, 121–136 (1975).
- Fukushima, K. — *A hierarchical neural network model for associative memory.* — Biol. Cybern. 50, 105–113 (1984).
- Fukushima, K., Miyake, S., Ito, T. — *Neocognitron: A neural network model for a mechanism of visual pattern recognition.* — IEEE Trans. SMC-13. 826–834 (1983).
- Fukushima, K. — *Neocognitron: A hierarchical neural network capable of visual pattern recognition.* — Neural Networks 1, 119–130 (1988).
- Gabriel, R. — *Massively parallel computers and NON-VON.* — Science 231, 975–978 (1986).
- Gardner, E. — *The space of interactions in neural network models.* — J. Phys. A: Math. Gen. 21, 257–270 (1988).
- Gardner, E. — *Maximum storage capacity in neural networks.* — Europhys. Lett. 4, 481–485 (1987).
- Gardner, E., Derrida, B., Mottishaw, P. — *Zero temperature parallel dynamics for infinite range spin glasses and neural networks.* — J. Phys., Paris 48, 741–755 (1987).
- Gardner, E., Stroud, N., Wallace, D.J. — *Training with noise and the storage of correlated patterns in a neural network model.* — Preprint, Edinburgh 87/394 (1987).
- Gardner, E. — *Multiconnected neural network models.* — J. Phys. A: Math. Gen. 20, 3453–3464 (1987).
- Gardner, E., Derrida, B. — *Optimal storage properties of neural network models.* — J. Phys. A: Math. Gen. 21, 271–284 (1988).

- Gelfand, A. — *A behavioral summary for completely random nets.* — Bull. Math. Biol. 44, 309–320 (1982).
- Gelperin, A., Hopfield, J.J. — *The logic of Limax learning.* — In 'Model Neural Networks and Behavior', Selverston, A. Ed., Plenum N.Y., pp. 237–261 (1985).
- Geman, S., Geman, D. — *Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images.* — IEEE Trans. PAMI 6, 721–741 (1984).
- Geszti, T., Pazmandi, F. — *Learning within bounds and dream sleep.* — J. Phys. A: Math. Gen. L1299–L1303 (1987).
- Geszti, T., Csabai, I. — *Boundary dynamics in the free and forced Kohonen map.* — Preprint, Eötvös University, Budapest, (1988).
- Getting, P. — *Reconstruction of small neural networks.* — In 'Methods in Neuronal Modeling: From Synapses to Networks', Koch, C., Segev, I. Eds., MIT Press, Cambridge, MA (1988).
- Gibbs, W.R. — *The eigenvalues of a deterministic neural net.* — Math. Biosci. 57, 19–34 (1981).
- Gilbert, C. — *Microcircuitry of the visual cortex.* — Ann. Rev. Neurosci. 6, 217–247 (1983).
- Giles, C.L., Maxwell, T. — *Learning, invariance, and generalization in high-order networks.* — Appl. Opt. 26, 4972–4978 (1987).
- Gill, M., Mullhaupt, A. — *Boolean delay equations. II. periodic and aperiodic solutions.* — J. Stat. Phys. 41, 125–173 (1985).
- Giraud, B., Axelrad, H., Liu, L. — *Flexibility of one-layer neural models.* — Preprint, Commissariat à l'Energie Atomique (Saclay, 1988).
- Gislén, L., Peterson, C., Söderberg, B. — *'Teachers and Classes' with neural networks.* — Preprint, University of Lund, Sweden, N° LU TP 89/19 (1989).
- Glauber, R. — *Time-dependent statistics of the Ising model.* — J. Math. Phys. 4, 294–307 (1963).
- Goelet, P., Castellucci, V.F., Schacher, S., Kandel, E.R. — *The long and the short of long-term memory — A molecular framework.* — Nature 322, 419–422 (1986).
- Goles, E., Fogelman, F., Pellegrin, D. — *Decreasing energy functions as a tool for studying threshold networks.* — Disc. Appl. Math. 12, 261–277 (1985).
- Goles, E. — *Ph. D. thesis.* — USMG/IMAG Grenoble Universities, France (1985).
- Golomb, D., Rubin, N., Sompolinsky, H. — *Willshaw model: Associative memory with sparse coding and low firing rates.* — Phys. Rev. A 41, 1843–1854 (1990).
- Gordon, M. — *Memory capacity of neural networks learning within bounds.* — J. Phys., Paris 48, 2053–2058 (1987).
- Gordon, M. — *Neural networks: Learning rules and memory capacity.* — In 'Systems with Learning and Memory Abilities', Delacour, J., Levy, J.C. Eds., Elsevier Amsterdam, pp. 465–482 (1988).
- Gordon, M., Peretto, P. — *The statistical distribution of Boolean gates in two-inputs, one-output multilayered neural networks.* — J. Phys. A: Math. Gen. 23, 3061–3072 (1990).

- Gorman, R., Sejnowski, T. — *Analysis of hidden units in a layered network trained to classify sonar targets.* — *Neural Networks.* 1, 75–89 (1988).
- Gouzé, J.L., Lasry, J.M., Changeux, J.P. — *Selective stabilization of muscle innervation during development: A mathematical model.* — *Biol. Cybern.* 46, 207–215 (1983).
- Graf, H.P., Jackel, R.E., Howard, R.E., Straughn, B., Denker, J.S., Hubbard, W., Tennant, D.M., Schwartz, D. — *VLSI implementation of a neural network memory with several hundreds of neurons.* — In Proc. 151 of the AIP Conference on ‘Neural Networks for Computing’, Snowbird, Utah, AIP (1986).
- Grant, M., Gunton, J. — *Cellular automata, Langevin equations and unstable states.* — *Phys. Rev. Lett.* 57, 1970–1973 (1986).
- Grassberger, P., Proccacia, I. — *Measuring the strangeness of strange attractors.* — *Physica* 9D, 189–208 (1983).
- Greville, T.N. — *Some applications of the pseudo-inverse of a matrix.* — *SIAM.* 2, 15–22 (1960).
- Grossman, T., Meir, R., Domany, E. — *Learning by choice of internal representations.* — *Complex Syst.* 2, 555 (1988).
- Grondin, R., Porod, W., Loeffler, C., Ferry, D. — *Synchronous and asynchronous systems of threshold elements.* — *Biol. Cybern.* 49, 1–7 (1983).
- Grossberg, S. — *Neural pattern discrimination.* — *J. Theor. Biol.* 27, 291–337 (1970).
- Grossberg, S. — *On the dynamics of operant conditioning.* — *J. Theor. Biol.* 33, 225–255 (1971).
- Grossberg, S. — *A neural theory of punishment and avoidance. I. Qualitative theory.* — *Math. Biosci.* 15, 39–67 (1972).
- Grossberg, S. — *A neural theory of punishment and avoidance. II. Quantitative theory.* — *Math. Biosci.* 15, 253–285 (1972).
- Grossberg, S. — *Classical and instrumental learning by neural networks.* — In ‘Progress in Theoretical Biology’, Rosen, R., Snell, F. Eds., Academic Press N.Y. (1974).
- Grossberg, S. — *A neural model of attention, reinforcement, and discrimination learning.* — *Int. Rev. Neurobiol.* 18, 263–327 (1975).
- Grossberg, S. — *A theory of human memory: Self-organization and performance of sensory-motor codes, maps, and plans.* — *Prog. Theor. Biol.* 5, 233–374 (1978).
- Gutfreund, H. — *Neural networks with hierarchically correlated patterns.* — *Phys. Rev. A* 37, 570–577 (1988).
- Gutfreund, H., Mézard, M. — *Processing of temporal sequences in neural networks.* — *Phys. Rev. Lett.* 61, 235–238 (1988).
- Gutschow, T. — *Neurocomputers find the pattern.* — *Electronic Syst. Design Mag.*, 57–62 (October 1988).
- Hampson, S., Kibler, D. — *A Boolean complete neural model of adaptive behavior.* — *Biol. Cybern.* 49, 9–19 (1983).

- Hampson, S.E., Volper, D.J. — *Linear function neurons: Structure and training.* — Biol. Cybern. 53, 203–217 (1986).
- Hampson, S.E., Volper, D.J. — *Disjunctive models of Boolean category learning.* — Biol. Cybern. 56, 121–137 (1987).
- Hansel, D., Sompolinsky, H. — *Learning from examples in a single layer neural network.* — Preprint, Racah Institute of Physics, Jerusalem (1989).
- Hartmann, G. — *Recursive features of circular receptive fields.* — Biol. Cybern. 43, 199–208 (1982).
- Hartmann, G. — *Processing of continuous lines and edges by the visual system.* — Biol. Cybern. 47, 43–50 (1983).
- Hastings, H., Pekelney, R. — *Stochastic information processing in biological systems.* — Biosyst. 15, 155–168 (1982).
- Hastings, H., Waner, S. — *Principles of evolutionary learning design for a stochastic neural network.* — Biosyst. 18, 105–109 (1985).
- Häussler, A., Von der Malsburg, C. — *Development of retinotopic projections: An analytical treatment.* — J. Theor. Neurobiol. 2, 47–73 (1983).
- Hawkins, R.D., Kandel, E. — *Is there a cell-biological alphabet for simple forms of learning?* — Psychol. Rev. 91, 375–391 (1984).
- Heidmann, A., Heidmann, T., Changeux, J.P. — *Stabilisation sélective de représentations neuronales par résonance entre 'préreprésentations' spontanées de réseau cérébral et 'percepts' évoqués par interaction avec le monde extérieur.* — C. R. Acad. Sci. Paris 299, 839–844 (1984).
- Heidmann, T., Changeux, J.P. — *Un modèle moléculaire de régulation d'efficacité au niveau post-synaptique d'une synapse chimique.* — C. R. Acad. Sci. Paris 295, 665–670 (1984).
- Hendrickson, A. — *Dots, stripes, and columns in monkey visual cortex.* — Trends Neurosci. TINS 406–410 (September 1985).
- Hecht-Nielsen, R. — *Applications of counterpropagation networks.* — Neural Networks 1, 131–139 (1988).
- Hertz, J.A., Thorbergsson, G.I. — *Dynamics of learning in simple perceptrons.* — Preprint, Nordita, Copenhagen, N° 88/20 S (1988).
- Hertz, J.A., Thorbergsson, G.I., Krogh, A. — *Phase transitions in simple learning.* — Preprint, Nordita, Copenhagen, N° 89/6 S (1989).
- Hérault, J. — *Le traitement de l'information dans les structures nerveuses.* — Ph. D. thesis, Univ. Scientifique et Médicale de Grenoble (1980).
- Hermann, H. — *Fast algorithm for the simulation of Ising models.* — J. Stat. Phys. 45, 145–151 (1986).
- Herz, A., Kühn, R., van Heinmen, J. — *Hebbian learning reconsidered: Representation of static and dynamic objects in associative neural nets.* — Preprint, Heidelberg 463 (1988).
- Hillis, D. — *The Connection Machine: A computer architecture based on cellular automata.* — Physica 10D, 213–228 (1984).
- Hinton, G., Sejnowski, T., Ackley, D. — *Boltzmann machines: Constraint satisfaction networks that learn.* — Carnegie-Mellon University, Technical Report CMU-CS-84-119, Pittsburgh PA. (1984).

- Hinton, G., Sejnowski, T. — *Optimal perceptual inference*. — IEEE Conference on 'Computer Vision and Pattern Recognition', pp. 448-453 (1983).
- Hodgkin, A.L., Huxley, A.F. — *A quantitative description of membrane current and its application to conduction and excitation in nerve*. — J. Physiol., London 117, 500-544 (1952).
- Hogg, T., Huberman, B.A. — *Understanding biological computation: Reliable learning and recognition*. — Proc. Natl Acad. Sci. USA 81, 6871-6875 (1984).
- Hogg, T., Huberman, B.A. — *Attractors on finite sets: The dissipative dynamics of computing structures*. — Phys. Rev. A32, 2338-2346 (1985).
- Hogg, T., Huberman, B.A. — *Parallel computing structures capable of flexible associations and recognitions of fuzzy inputs*. — J. Stat. Phys. 41, 115-123 (1985).
- Holden, A. — *Stochastic processes in neurophysiology: Transformation from point to continuous processes*. — Bull. Math. Biol. 45, 443-465 (1983).
- Hopfield, J.J. — *Neural networks and physical systems with emergent collective computational abilities*. — Proc. Natl Acad. Sci. USA 79, 2554-2558 (1982).
- Hopfield, J.J. — *Neurons with graded response have collective computational properties like those of two-state neurons*. — Proc. Natl Acad. Sci. USA 81, 3088-3092 (1984).
- Hopfield, J.J. — *Computing with neural circuits: A model*. — Science 233, 625-632 (1986).
- Hopfield, J.J., Feinstein, D.I., Palmer, R.G. — *'Unlearning' has a stabilizing effect in collective memories*. — Nature 304, 158-159 (1983).
- Hopfield, J.J., Tank, D.W. — *'Neural' computation of decisions in optimization problems*. — Biol. Cybern. 52, 141-152 (1985).
- Hopfield, J.J. — *Learning algorithms and probability distributions in feed-forward and feedback networks*. — Proc. Natl Acad. Sci. USA 84, 8429-8433 (1987).
- Horn, D., Weyers, J. — *Information packing in associative memory models*. — Phys. Rev. A 34, 2324-2328 (1986).
- Horn, D., Usher, M. — *Neural networks with dynamical thresholds*. — Phys. Rev. A 40, 1036-1044 (1989).
- Hubel, D.H., Wiesel, T.N. — *Receptive fields binocular interaction and functional architecture in the cat's visual cortex*. — J. Physiol., London 160, 106-154 (1962).
- Hubel, D.H., Wiesel, T.N. — *Functional architecture of macaque monkey visual cortex*. — Proc. R. Soc., London B 198, 1-59 (1977).
- Hubel, D., Wiesel, T.N., Stryker, M.P. — *Anatomical demonstration of orientation columns in macaque monkey*. — J. Comparative Neurol. 177, 361-379 (1978).
- Huberman, B., Hogg, T. — *Phase transitions in Artificial Intelligence systems*. — Art. Intell. 33, 155-171 (1987).
- Imbert, M. — *La neurobiologie de l'image*. — La Recherche 144, 600-613 (1983).

- Ingber, L. — *Statistical mechanics of neocortical interactions 1: Basic formulation.* — Physica D 5, 83–107 (1982) and Phys. Rev. A 28, 395–416 (1983).
- Ito, M. — *Questions in modeling the cerebellum.* — J. Theor. Biol. 99, 81–86 (1982).
- Itoh, K. — *A neuro-synaptic model of the auditory masking and unmasking process.* — Biol. Cybern. 52, 229–235 (1985).
- Iversen, L. — *Chemical to think by.* — New Scientist 11–14 (May 1985).
- Jackel, L.D., Howard, R.E., Graf, H.P., Straughn, B., Denker, J.S. — *Artificial neural networks for computing.* — J. Vac. Sci. Technol. B4, 61–63 (1986).
- Jackel, L.D., Graf, H.P., Howard, R.E. — *Electronic neural network chips.* — Appl. Opt. 26, 5077–5080 (1987).
- Jackel, L., Howard, R., Denker, J., Hubbard, W., Solla, S. — *Building a hierarchy with neural networks: An example-image vector quantization.* — Appl. Opt. 26, 5081–5084 (1987).
- Jacobs, R.A. — *Increased rates of convergence through learning rate adaptation.* — Neural Networks 1, 295–307 (1988).
- Jeffrey, W., Rosner, R. — *Optimization algorithms: Simulated annealing and neural network processing.* — Astrophys. J. 310, 473–481 (1986).
- John, R. — *Switchboard versus statistical theories of learning and memory.* — Science 177, 850–864 (1972).
- Johnston, V., Partridge, D., Lopez, P. — *A neural theory of cognitive development.* — J. Theor. Biol. 100, 485–509 (1983).
- Jones, E.G. — *Anatomy of cerebral cortex: Columnar organization.* — In ‘The Organization of the Cerebral Cortex’, Schmitt, F., Worden, F., Adelman, G., Dennis, S. Eds., p. 199, MIT Press, Cambridge, MA (1981).
- Jones, W.P., Hoskins, J. — *Back-propagation: A generalized delta learning rule.* — BYTE, 155–162 (October 1987).
- Jones, R., Lee, Y., Barnes, C., Flake, G., Lee, K., Lewis, P., Qian, S. — *Function approximation and time series prediction with neural networks.* — Preprint, University of California (1989).
- Josin, G. — *Neural-network heuristics: Three heuristic algorithms that learn from experience.* — BYTE, 183–192 (October 1987).
- Judd, S. — *On the complexity of loading shallow neural networks.* — J. Complexity 4, 177–192 (1988).
- Julesz, B. — *Experiments in the visual perception of texture.* — Sci. Am. 232, 34–43 (1975).
- Julesz, B. — *A theory of preattentive texture discrimination based on first-order statistics of textons.* — Biol. Cybern. 41, 431–138 (1981).
- Julesz, B. — *Textons, the elements of texture perception, and their interactions.* — Nature 209, 91–97 (1981).
- Kandel, E.R. — *Small systems of neurons.* — Sci. Am. 241, 66–76 (1979).

- Kandel, E., Tauc, L. — *Mechanism of heterosynaptic facilitation in the giant cell of abdominal ganglion of Aplysia depilans.* — J. Physiol (London) 181, 28–47 (1965).
- Kanter, I., Sompolinsky, H. — *Associative recall of memory without errors.* — Phys. Rev. A 35, 380–392 (1987).
- Kanter, I., Sompolinsky, H. — *Mean-field theory of spin glasses with finite coordination number.* — Phys. Rev. Lett. 58, 164–167 (1987).
- Karp, R., Pearl, J. — *Searching for an optimal path in a tree with random costs.* — Art. Intell. 21, 99–116 (1983).
- Kauffman, S.A. — *Metabolic stability and epigenesis in randomly constructed genetic nets.* — J. Theor. Biol. 22, 437–467 (1969).
- Kauffman, S.A. — *Behavior of randomly constructed genetic nets.* — In ‘Towards a Theoretical Biology’, vol. 3, Waddington, C.H. Ed., Edinburgh Univ. Press, pp. 18–37 (1970).
- Kauffman, S.A. — *Emergent properties in random complex automata.* — Physica D 10, 145–156 (1984).
- Katz, B., Miledi, R. — *A study of synaptic transmission in the absence of nerve impulses.* — J. Physiol. 192, 407–436 (1967).
- Kawasaki, K. — *Kinetics of Ising models.* — In ‘Phase Transitions and Critical Phenomena’, vol. II, Domb, C., Grenn, M.S. Eds., Academic Press, pp. 443–501 (1972).
- Kennedy, M., Chua, L. — *Unifying the Tank and Hopfield linear programming circuit and the canonical nonlinear programming circuit of Chua and Lin.* — IEEE Trans. CAS-34, 210–214 (1987).
- Kennedy, M., Chua, L. — *Neural networks for nonlinear programming.* — IEEE Trans. CS-35, 554–562 (1988).
- Kepler, T., Abbott, L. — *Domains of attraction in neural networks.* — J. Phys., Paris 49, 1657–1662 (1988).
- Kerszberg, M., Mukamel, D. — *Dynamics of simple computer networks.* — J. Stat. Phys. 51, 777–795 (1988).
- Kerszberg, M., Zippelius, A. — *Synchronization in neural assemblies.* — Preprint, IFK, Jülich (1988).
- Kienker, P., Sejnowski, T., Hinton, G., Schumacher, L. — *Separating figure from ground with a parallel network.* — Perception 15, 197–216 (1986).
- Kim, M., Hsu, C. — *Computation of the largest Lyapunov exponent by the generalized cell mapping.* — J. Stat. Phys. 45, 49–61 (1986).
- Kinoshita, J. — ‘*Neural Darwinism*’: Competition among neurons is simulated on computers. — Sci. Am. 258, 11–12 (1988).
- Kinzel, W. — *Phase transition of cellular automata.* — Z. Phys. B 58, 229–244 (1985).
- Kinzel, W. — *Models of neural networks.* — In Proc. Gwatt meeting: ‘Physics and the Living Matter’, Lecture Notes in Physics, Springer, Berlin (1987).
- Kinzel, W. — *Learning and pattern recognition in spin-glass models.* — Z. Phys. B 60, 205–213 (1985).

- Kinzel, W. — *Remanent magnetization of the infinite-range Ising spin glass.* — *Phys. Rev. B* 33, 5086–5088 (1986).
- Kinzel, W. — *Neural networks with asymmetric bonds.* — In Proc. 1986 Heidelberg Coll. ‘Glassy dynamics and Optimization’, Van Hemmen, L.V., Morgenstern, I., Eds., Springer, Berlin (1987).
- Kirilov, A., Borisuk, G., Borisuk, R., Kovalenko, Y., Kryukov, V. — *Short-term memory as a metastable state. III: Diffusion approximation.* — *Cybern. Syst.* 17, 169–182 (1986).
- Kirkpatrick, S., Gelatt, C., Vecchi, M. — *Optimization by simulated annealing.* — *Science* 220, 671–680 (1983).
- Kirkpatrick, S., Sherrington, D. — *Infinite-range models of spin-glasses.* — *Phys. Rev. B* 17, 4384–4403 (1978).
- Kitatsuji, Y., Kuroda, T. — *Statistical study on reverberation between two mutually excitatory neuron groups.* — *J. Theor. Biol.* 100, 25–55 (1983).
- Kleinfeld, D. — *Sequential state generation by model neural networks.* — *Proc. Natl Acad. Sci. USA* 83, 9469–9473 (1986).
- Kleinfeld, D., Raccuia, G.F., Chiel, H.J. — *Circuits with bistable outputs constructed from identified *Aplysia* neurons.* — Preprint, AT & T Bell Laboratories, Murray Hill, N.J. (1988).
- Kleinfeld, D., Sompolinsky, H. — *Associative neural network model for the generation of temporal patterns: Theory and application to Central Pattern Generators.* — *Biophys. J.* 54, 1039–1051 (1988).
- Kloeden, P., Mees, A. — *Chaotic phenomena.* — *Bull. Math. Biol.* 47, 697–738 (1985).
- Knowles, D., Traub, R., Wong, R., Miles, R. — *Properties of neural networks: Experimentation and modeling of the epileptic hippocampal slice.* — *Trends Neurosci.*, 73–79 (February 1985).
- Koch, C., Marroquin, J., Yuille, A. — *Analog ‘neuronal’ networks in early vision.* — *Proc. Natl Acad. Sci. USA* 83, 4263–4267 (1986).
- Koenderink, J. — *Simultaneous order in nervous nets from a functional standpoint.* — *Biol. Cybern.* 50, 35–41 (1984).
- Koenderink, J. — *Geometrical structures determined by the functional order in nervous nets.* — *Biol. Cybern.* 50, 43–50 (1984).
- Koenderink, J. — *The structure of images.* — *Biol. Cybern.* 50, 363–370 (1984).
- Kohonen, T. — *Correlation matrix memories.* — *IEEE Trans. C-21*, 353–359 (1972).
- Kohonen, T., Ruohonen, M. — *Representation of associated data by matrix operators.* — *IEEE Trans. C-22*, 701–702 (1973).
- Kohonen, T. — *Analysis of a simple self-organizing process.* — *Biol. Cybern.* 44, 135–140 (1982).
- Kohonen, T. — *Self-organized formation of topologically correct feature maps.* — *Biol. Cybern.* 43, 59–69 (1982).
- Kohonen, T., Oja, E. — *Fast adaptive formation of orthogonalizing filters and associative memory in recurrent networks of neuron like elements.* — *Biol. Cybern.* 21, 85–95 (1976).

- Kohonen, T. — *An adaptive associative memory principle*. — IEEE Trans. C-23 444–445 (1974).
- Kohonen, T. — *Automatic formation of topological maps of patterns in a self-organizing system*. — In the Proceeding of the '2nd Scandinavian Conference on Image Analysis', Oja, E., Simula, O. Eds., Seura, Espoo (Finland), pp. 214–220 (1981).
- Kohonen, T. — *Analysis of a simple self-organizing process*. — Biol. Cybern. 44, 135–140 (1982).
- Kohonen, T., Lehtio, P., Rovamo, J., Hyvärinen, J., Bry, K. — *A principle of neural associative memory*. — Neurosci. 2, 1065–1076 (1977).
- Kohonen, T., Reuhkala, E., Mäkisara, K., Vainio, L. — *Associative recall of images*. — Biol. Cybern. 22, 159–168 (1976).
- Kohonen, T. — *Adaptive, associative, and self-organizing functions in neural computing*. — Appl. Opt. 26, 4910–4918 (1987).
- Kolmogorov, A.N. — *On the representation of continuous functions of several variables by superposition of continuous functions of one variable and addition*. — Dokl. Akad. Nauk. (SSSR) 114, 953–956 (1957) (in Russian); Am. Math. Soc. Transl. 2, 55–59 (1963).
- Kosko, B. — *Counting with fuzzy sets*. — IEEE Trans. PAMI-8, 556–557 (1986).
- Kosko, B. — *Fuzzy entropy and conditioning*. — Inform. Sci. 40, 1–10 (1987).
- Kovbasa, S.I. — *The metalanguage of neuron groups*. — J. Theor. Neurobiol. 4, 153–178 (1985).
- Krauth, W., Mézard, M. — *Learning algorithms with optimal stability in neural networks*. — J. Phys. A: Math. Gen. 20, L745–L751 (1987).
- Krauth, W., Nadal, J.P., Mézard, M. — *The roles of stability and symmetry in the dynamics of neural networks*. — J. Phys. A: Math. Gen. 21, 2995–3011 (1988).
- Krauth, W., Mézard, M., Nadal, J.P. — *Basins of attraction in a perceptron-like neural network*. — Complex Syst. 2, 387–408 (1988).
- Kree, R., Zippelius, A. — *Continuous time dynamics of asymmetrically diluted neural networks*. — Phys. Rev. A 36, 4421–4427 (1987).
- Kree, R., Zippelius, A. — *Recognition of topological features of graphs and images in neural networks*. — J. Phys. A 21, L813–L818 (1988).
- Kree, R., Widmaier, D., Zippelius, A. — *Spin glass phase in a neural network with asymmetric couplings*. — J. Phys. A: Math. Gen. 21, L1181–L1186 (1988).
- Krogh, A., Hertz, J. — *Hebbian learning of principal components*. — Preprint, Nordita, Copenhagen, N° 89/50 S (1989).
- Krogh, A., Hertz, J.A., Thorbergsson, G.I. — *A cost function for internal representations*. — Preprint, Nordita Copenhagen, N° 89/37 S (1989).
- Krogh, A., Hertz, J.A. — *Mean-field analysis of hierarchical associative networks with 'magnetization'*. — Preprint, Niels-Bohr Institute, Copenhagen, to appear in J. Phys. A (1989).
- Krüger, J. — *Simultaneous individual recordings from many cerebral neurons: Techniques and results*. — Rev. Phys., Bio., Pharm. 98, 177–233 (1983).

- Krüger, J., Bach, M. — *Independent systems of orientation columns in upper and lower layers of monkey visual cortex.* — *Neurosci. Lett.* 31, 225–230 (1982).
- Kryukov, V.I. — *Markov interaction processes and neural activity.* — *Lect. Notes Math.* 653, 122–139 (1978).
- Kryukov, V.I. and Kovalenko, Y., Borisuk, R., Borisuk, G., Kirilov, A., Kryukov, V.I. — *Short-term memory as a metastable state. I: Master equation approach, II: Simulation model.* — In 'Cybernetics and System Research 2', Trappi, R. Ed., Elsevier Amsterdam, pp. 261–271 (1984).
- Kupperstein, M. — *Neural model of adaptive hand-eye coordination for single postures.* — *Science* 239, 1308–1311 (1988).
- Kurka, P. — *Markov chains with infinite transition rates.* — *Math. Biosci.* 62, 137–149 (1982).
- Kürten, K.E., Clark, J.W. — *Chaos in neural systems.* — *Phys. Lett.* 114 A, 413–418 (1986).
- Kürten, K. — *Critical phenomena in model neural networks.* — *Phys. Lett. A* 129, 157–160 (1988).
- Kürten, K. — *Correspondance between neural threshold networks and Kauffman Boolean cellular automata.* — *J. Phys. A: Math. Gen.* 21, L615–L619 (1988).
- Kürten, K. — *Dynamical phase transitions in short-ranged and long-ranged neural networks models.* — *J. Phys., Paris* 50, 2313–2323 (1989).
- Kushnir, M., Abe, K., Matsumoto, K. — *Recognition of handprinted hebrew characters using features selected in the Hough transform space.* — *Patt. Recog.* 18, 103–114 (1985).
- Lallemand, P., Diep, H.T., Ghazali, A., Toulouse, G. — *Configuration space analysis for fully frustrated vector spins.* — *J. Physique Lett.* 46, L1087–L1093 (1985).
- Lang, K.J., Witbrock, M.J. — *Learning to tell two spirals apart.* — In Proc. 'Connectionist Summer School', Touretzky, D., Hinton, G., Sejnowski, T. Eds., Morgan Kaufmann (1988).
- Lansky, P. — *On approximation of Stein's neuronal model.* — *J. Theor. Biol.* 107, 631–647 (1984).
- Lapedes, A., Farber, R. — *Non-linear signal processing using neural networks: Prediction and system modeling.* — Report LA-UR 87-545, Los Alamos National Laboratory, NM (1987).
- Lapedes, A., Farber, R. — *A self-organizing, non-symmetrical neural net for content addressable memory and pattern recognition.* — *Physica, Amsterdam D* 22, 247–259 (1987).
- Lapedes, A. — *How neural nets work.* — In 'Neural Information Processing Systems', Anderson, D.Z. Ed., AIP, pp. 442–456 (1988).
- Larrabee, M.G., Bronk, P.W. — *Prolonged facilitation of synaptic excitation in sympathetic ganglia.* — *J. Neurophysiol* 10, 139–154 (1947).
- Larson, J., Lynch, G. — *Induction of synaptic potentiation in hippocampus by patterned stimulations involves two events.* — *Science* 232, 985–988 (1986).

- Law, M.I., Constantine-Paton, M. — *Anatomy and physiology of experimentally produced striped tecta*. — J. Neurosci. 1, 741–759 (1981).
- Le Cun, Y. — *A learning scheme for asymmetric threshold network*. — In 'Cognitiva 85', Cesta-AFCET Eds., Paris, 599–604 (1985).
- Le Cun, Y., Guyon, I., Jackel, L., Henderson, D., Boser, B., Howard, R., Denker, J., Hubbard, W., Graf, H. — *Handwritten digit recognition: Applications of neural network chips and automatic learning*. — IEEE Commns Mag. 41–46 (November 1989).
- Lehky, S., Sejnowski, T. — *Network model of shape-from-shading: Neural function arises from both receptive and projective fields*. — Nature 333, 452–454 (1988).
- Lettvin, T.Y., Maturana, H.R., McCulloch, W.S., Pitts, W. — *What the frog's eye tells to the frog's brain*. — Proc. IRE. 47, 1940–1951 (1959).
- Levay, S. — *Synaptic patterns in the visual cortex of the cat and monkey. Electron microscopy of Golgi preparations*. — J. Comp. Neurol. 150, 53–86 (1973).
- Levenstein, M., Nowak, A. — *Fully connected neural networks with self-control of noise levels*. — Phys. Rev. Lett. 62, 225–228 (1989).
- Levin, E., Tishby, N., Solla, S. — *A statistical approach to learning and generalization in layered neural networks*. — In Proc. workshop 'Computational Learning Theory', COLT'89 (1989).
- Levine, D. — *Neural population modeling and psychology: A review*. — Math. Biosci. 66, 1–86 (1983).
- Levy, M.B. — *Associative changes at the synapse: LTP in the hippocampus*. — In 'Synaptic Modification, Neuron Selectivity, and Nervous System Organization', Levy, W., Anderson, J., Lehmkuhle, S. Eds., Lawrence Erlbaum, London, p. 5 (1985).
- Levy, W., Desmond, N. — *The rules of elemental synaptic plasticity*. — In 'Synaptic Modification, Neuron Selectivity, and Nervous System Organization', Levy, W., Anderson, J., Lehmkuhle, S. Eds., Erlbaum, Hillsdale, NJ, pp. 105–121 (1985).
- Levy, W., Steward, O. — *Synapses as associative memory elements in the hippocampal formation*. — Brain Res. 175, 233–245 (1979).
- Lin, S., Kernighan, B.W. — *An effective heuristic algorithm for the traveling-salesman problem*. — Oper. Res. 21, 498–516 (1973).
- Linsker, R. — *From basic network principles to neural architecture 1: Emergence of spatial-opponent cells*. — Proc. Natl Acad. Sci. USA 83, 7508–7512 (1986). — *2: Emergence of orientation-selective cells*. — Proc. Natl Acad. Sci. USA 83, 8390–8394 (1986). — *3: Emergence of orientation columns*. — Proc. Natl Acad. Sci. USA 83, 8779–8783 (1986).
- Lippmann, R. — *An introduction to computing with neural nets*. — IEEE ASSP Mag. 4–22 (April 1987).
- Lippmann, R., Gold, B., Malpass, M. — *A comparison of Hamming and Hopfield neural nets for pattern classification*. — MIT Technical Report 769 (May 1987).

- Lippmann, R. — *An introduction to computing with neural nets.* — IEEE ASSP Mag. 4, 4-22 (April 1987).
- Lippmann, R. — *Review of neural networks for speech recognition.* — Neural Comp. 1, 1-38 (1989).
- Lippmann, R. — *Pattern classification using neural networks.* — IEEE Commns Mag. 47-64 (November 1989).
- Little, W.A. — *The existence of persistent states in the brain.* — Math. Biosci. 19, 101-120 (1974).
- Little, W.A., Shaw, G.L. — *A statistical theory of short and long term memory.* — Behav. Biol. 14, 115-133 (1975).
- Little, W.A., Shaw, G.L. — *Analytic study of the memory storage capacity of a neural network.* — Math. Biosci. 39, 281-290 (1978).
- Livingstone, M. — *Art, illusion, and the visual system.* — Sci. Am. 258, 68-75 (1988).
- Livingstone, M., Hubel, D. — *Segregation of form, color, movement, and depth: Anatomy, physiology, and perception.* — Science 240, 740-749 (1988).
- Llinas, R., Sugimori, M., Simon, S.M. — *Transmission by presynaptic spikelike depolarization in the squid giant synapse.* — Proc. Natl Acad. Sci. USA 79, 2415-2419 (1982).
- Lloyd, D.P.C. — *Post tetanic potentiation of response in monosynaptic reflex pathways of the spinal chord.* — J. Gen. Physiol. 33, 147-170 (1949).
- Lloyd, D.P.C., Wilson, V.J. — *Reflex depression in rhythmically active monosynaptic reflex pathways.* — J. Gen. Physiol. 40, 409-426 (1957).
- Longuet-Higgins, H.C. — *Holographic model of temporal recall.* — Nature 217, 104 (1968).
- Lorentz, G. — *Metric entropy, widths, and superpositions of functions.* — Am. Math. Monthly 69, 469-485 (1962).
- Lupo, J. — *Defense applications of neural networks.* — IEEE Commns Mag. 82-88 (November 1989).
- Lynch, G., Baudry, M. — *The biochemistry of memory: A new and specific hypothesis.* — Science 224, 1057-1063 (1984).
- Marcus, C., Waugh, F., Westervelt, R. — *Associative memory in an analog iterated-map neural network.* — Phys. Rev. A 41, 3355-3364 (1990).
- Marr, D. — *A theory of the cerebellar cortex.* — J. Physiol. 202, 437-470 (1969).
- Marr, D. — *A theory of cerebral neocortex.* — Proc. Roy. Soc. London. B 176, 161-234 (1970).
- Marr, D., Poggio, T. — *Cooperative computation of stereo disparity.* — Science 194, 283-287 (1976).
- Mattis, P.C. — *Solvable spin systems with random interactions.* — J. Physique Lett. A 56, 421-422 (1976).
- Maxwell, T., Giles, C. — *Transformation invariance using high order correlations in neural net architectures.* — In Proceedings IEEE Int. Conf. Syst., Man and Cybernetics, Atlanta, Georgia, pp. 627-632 (October 1986).

- McCulloch, W.S., Pitts, W. — *A logical calculus of the ideas immanent in nervous activity.* — Bull. Math. Biophys. 5, 115–133 (1943).
- McDermott, J. — *Face to face: It's the expression that bears the message.* — Smithsonian 16, 112–123 (March 1986).
- McEliece, R., Posner, E., Rodemich, E., Venkatesh, S. — *The capacity of the Hopfield associative memory.* — IEEE Trans. IT-33, 461–482 (1987).
- Meir, R., Domany, E. — *Iterated learning in a layered feedforward neural network.* — Phys. Rev. A 37, 2660–2668 (1988).
- Meir, R. — *Extensions of a solvable feedforward neural network.* — J. Phys., Paris 49, 201–213 (1988).
- Meir, R., Domany, E. — *Stochastic dynamics of a layered neural network: Exact solution.* — Europhys. Lett. 4, 645–650 (1987).
- Meir, R., Domany, E. — *Exact solution of a layered neural network model.* — Phys. Rev. Lett. 59, 359–362 (1987).
- Meir, R., Domany, E. — *A layered feedforward neural network with exactly soluble dynamics.* — Phys. Rev. A 37, 608–618 (1988).
- Meunier, J., Cavanagh, P. — *Efficacité de la mémoire associative inhérente à la potentiation post-tétanique.* — Biol. Cybern. 53, 159–171 (1986).
- Meunier, C., Hansel, D., Verga, A. — *Information processing in three-state neural networks.* — Preprint, École Polytechnique, Palaiseau, France (1988).
- Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., Teller, E. — *Equation of state calculations by fast computing machines.* — J. Chem. Phys. 21, 1087–1092 (1953).
- Mézard, M. — *The space of interactions in neural networks: Gardner's computation with the cavity method.* — J. Phys. A: Math. Gen. 22, 2181–2190 (1989).
- Mézard, M., Parisi, G., Sourlas, N., Toulouse, G., Virasoro, M. — *Replica symmetry breaking and the nature of the spin glass phase.* — J. Phys. 45, 843–854 (1984).
- Mézard, M., Parisi, G., Virasoro, M. — *SK model: The replica solution without replicas.* — Europhys. Lett. 1, 77–82 (1986).
- Mézard, M., Nadal, J.P., Toulouse, G. — *Solvable models of working memories.* — J. Phys., Paris 47, 1457–1462 (1986).
- Mézard, M., Nadal, J.P. — *Learning in feedforward layered networks: The tiling algorithm.* — J. Phys. A: Math. Gen. 22, 2191–2203 (1989).
- Mézard, M., Patarnello, S. — *On the capacity of feedforward layered networks.* — Preprint, École Normale Supérieure, Paris, N° 89/24 (1989).
- Milgram, M., Atlan, H. — *Probabilistic automata as a model for epigenesis of cellular networks.* — J. Theor. Biol. 103, 523–547 (1983).
- Miller, G. — *The magical number seven, plus or minus two: Some limit in our capacity for processing information.* — Psychol. Rev. 63, 81–97 (1956).
- Minsky, M. — *Steps toward artificial intelligence.* — In 'Computers and Thought', Feigenbaum, E., Feldman, J. Eds., McGraw Hill, N.Y. (1963).
- Mitchinson, G.J., Durbin, R.M. — *Bounds on the learning capacity of some multi-layer networks.* — Research Memo, King's College, Cambridge, (1989).

- Miyake, S., Fukushima, K. — *A neural network model for the mechanism of feature-extraction: A self-organizing network with feedback inhibition.* — Biol. Cybern. 50, 377–384 (1984).
- Miyashita, Y., Chang, H.S. — *Neuronal correlate of pictorial short-term memory in the primate temporal cortex.* — Nature 331, 68–70 (1988).
- Moody, J., Darken, C. — *Fast learning in networks of locally tuned processing units.* — Neural Computation 1, 281–294 (1989).
- Moskowitz, N., Puszkin, S. — *A unified theory of presynaptic chemical neurotransmission.* — J. Theor. Biol. 112, 513–534 (1985).
- Mountcastle, V.B. — *Modality and topographic properties of single neurons of cat's somatic sensory cortex.* — J. Neurophys. 20, 408–434 (1957).
- Mountcastle, V.B. — *An organizing principle for cerebral function: The unit module and the distributed system.* — In 'The Mindful Brain', Schmitt, F. Ed., MIT Press, Cambridge, MA, pp. 7–50 (1978).
- Munro, P. — *A model for generalization and specification by single neuron.* — Biol. Cybern. 51, 169–179 (1984).
- Murdock, B. Jr. — *The serial position effect of free recall.* — J. Exp. Psychology 64, 482–488 (1962).
- Muroga, S., Toda, I., Takasu, S. — *Theory of majority decision elements.* — J. Franklin Inst. 271, 376–418 (May 1961).
- Nadal, J.P., Toulouse, G., Changeux, J.P., Dehaene, S. — *Networks of formal neurons and memory palimpsests.* — Europhys. Lett. 1, 535–542 (1986).
- Nadal, J.P., Toulouse, G. — *Information storage in sparsely-coded memory nets.* — Preprint, École Normale Supérieure, Paris (May 1989).
- Nakano, K. — *Associatron. A model of associative memory.* — IEEE Trans. SMC-2, 380–388 (1972).
- Nass, M., Cooper, L. — *A theory for the development of feature detecting cells in visual cortex.* — Biol. Cybern. 19, 252–264 (1975).
- Nelson, T. — *A neural network model for cognitive activity.* — Biol. Cybern. 49, 79–88 (1983).
- Nemeth, R. — *Sensibility of neural networks.* — J. Phys. A: Math. Gen. 20, L85–L88 (1987).
- Niebur, E., Erdös, P. — *Computer simulation of electrotonic neural networks.* — In 'Computer Simulation in Brain Science', Cottrill, R. Ed., Cambridge University Press, Cambridge, pp. 148–163 (1988).
- Niebur, E., Erdös, P. — *Theory of the locomotion of nematodes. I: Control of the somatic motor neurones by interneurones, II: Dynamics of undulatory movement on a solid surface.* — Preprint, ITP University of Lausanne (April and August 1989).
- Ninio, J. — *Modèle de mémoire iconique localisée, répllicable et associative, utilisant le temps partagé et trois modes de communications neuronales.* — C. R. Acad. Sci. Paris 306, 545–550 (1988).

- Nishimori, H., Nakamura, T., Shiino, M. — *Retrieval of spatio-temporal sequence in asynchronous neural network.* — Phys. Rev. A 41, 3346–3354 (1990).
- Noest, A.J. — *Associative memory in sparse phasor neural networks.* — Europhys. Lett. 6, 169–474 (1988).
- Norman, M., Radcliffe, N., Richards, G., Smieja, F., Wallace, D.. — *Neural network applications in the Edinburgh Concurrent Supercomputer project.* — Preprint, Edinburgh 89/462 (1989).
- Nguyen, D., Widrow, B. — *The truck backer-upper: An example of self-learning in neural networks.* — In Proc. Joint Int. Conf. Neural Networks (JICNN), Washington, pp. II-357–363 (June 1989).
- Obermeier, K. — *Parallel processing: Side by side.* — BYTE, 275–283 (1988).
- O'Donohue, T., Millington, W., Handelman, G., Contreras, P., Chronwall, B. — *On the 50th anniversary of Dale's law: Multiple neurotransmitter neurons.* — Trends in Pharm. Sci., Elsevier, Amsterdam, 305–308 (August 1985).
- Ogawa, H. — *Labeled point pattern matching by fuzzy relaxation.* — Patt. Recog. 17, 569–573 (1984).
- Ogielski, A., Stein, D. — *Dynamics on ultrametric spaces.* — Phys. Rev. Lett. 55, 1634–1637 (1985).
- Oguztöreli, M., Caelli, T. — *An inverse problem in neural processing.* — Biol. Cybern. 53, 239–245 (1986).
- Oguztöreli, M., Steil, G., Caelli, T. — *Control mechanisms of a neural network.* — Biol. Cybern. 54, 21–28 (1986).
- Okajima, K. — *A mathematical model of the primary visual cortex and hypercolumn.* — Biol. Cybern. 54, 107–114 (1986).
- Oja, E. — *A simplified neuron model as a principal component analyzer.* — J. Math. Biol. 15, 267–273 (1982).
- Oja, E. — *On the convergence of an associative learning algorithm in the presence of noise.* — Int. J. Syst. Sci. 11, 629–640 (1980).
- Oonuki, M. — *Firing probability of a neural network: First-order differential equation.* — J. Theor. Biol. 58, 59–81 (1976).
- Oonuki, M. — *Macroscopic law for time and activity in a neural system.* — J. Theor. Biol. 94, 191–196 (1982).
- Oonuki, M. — *Extension of a differential equation to neurons with complicated postsynaptic potentials.* — J. Theor. Neurobiol. 4, 95–111 (1985).
- Opper, M. — *Learning in neural networks: Solvable dynamics.* — Europhys. Lett. 8, 389–392 (1989).
- Opper, M. — *Learning time of neural networks: An exact solution for a perceptron algorithm.* — Phys. Rev. A 38, 3824–3829 (1988).
- Paine, G. — *The development of Lagrangians for biological models.* — Bull. Math. Biol. 44, 749–760 (1982).
- Palm, G. — *On representation and approximation of nonlinear systems.* — Biol. Cybern. 31, 119–124 (1978).

- Palm, G. — *On representation and approximation of nonlinear systems. Part II: Discrete time.* — Biol. Cybern. 34, 49–52 (1979).
- Palm, G. — *On the storage capacity of an associative memory with randomly distributed elements.* — Biol. Cybern. 39, 125–135 (1981).
- Palm, G. — *On associative memory.* — Biol. Cybern. 36, 19–31 (1980).
- Palm, G. — *Towards a theory of cell assemblies.* — Biol. Cybern. 39, 181–194 (1981).
- Palm, G. — *Rules for synaptic changes and their relevance for the storage of information in the brain.* — In 'Cybernetics and Systems Research', R. Trappl Ed., pp. 277–280, North Holland, Amsterdam (1982).
- Palm, G., Bonhoeffer, T. — *Parallel processing for associative and neural networks.* — Biol. Cybern. 51, 201–204 (1984).
- Palmer, R. — *Broken ergodicity.* — Adv. Phys. 31, 669–735 (1983).
- Parga, N., Parisi, G., Virasoro, M.A. — *A numerical investigation of the overlap distribution among pure states in the spin glass phase.* — J. Physique Lett. 45, L1063–L1069 (1984).
- Parga, N., Virasoro, M. — *The ultrametric organization of memories in a neural network.* — J. Phys. 47, 1857–1864 (1986).
- Parisi, G. — *Infinite number of order parameters for spin-glasses.* — Phys. Rev. Lett. 43, 1754–1756 (1979).
- Parisi, G. — *The order parameter for spin glasses: A function on the interval 0–1.* — J. Phys. A: Math. Gen. 13, 1101–1112 (1980).
- Parisi, G. — *Magnetic properties of spin glasses in a new mean field theory.* — J. Phys. A: Math. Gen. 13, 1887–1895 (1980).
- Parisi, G. — *Order parameter for spin glasses.* — Phys. Rev. Lett. 50, 1946–1948 (1983).
- Parisi, G. — *Asymmetric neural networks and the process of learning.* — J. Phys. A 19, L675–L680 (1986).
- Parisi, G. — *A memory which forgets.* — J. Phys. A: Math. Gen. 19, L617–L620 (1986).
- Parker, D. — *Learning-logic.* — Invention Report, S81-64, Office of Technology Licensing, Stanford University (October 1982).
- Patarnello, S., Carnevali, P. — *Learning Networks of Neurons with Boolean Logic.* — Europhys. Lett. 4, 503–508 (1987).
- Paulin, M.G. — *Is the cerebellum an adaptive filter?* — J. Theor. Neurobiol. 4, 143–151 (1985).
- Pauwelusen, J. — *One way traffic of pulses in a neuron.* — J. Math. Biol. 15, 151–171 (1982).
- Pearson, J., Finkel, L., Edelman, G. — *Plasticity in the organization of adult cerebral cortical maps: A computer simulation based on neuronal group selection.* — J. Neurosci. 7, 4209–4223 (1987).
- Pellioniz, A., Llinas, R. — *Tensorial approach to the geometry of brain function: Cerebellar coordination via a metric tensor.* — Neurosci. 5, 1125–1136 (1979).
- Peretto, P. — *Collective properties of neural networks: A statistical physics approach.* — Biol. Cybern. 50, 51–62 (1984).

- Peretto, P., Niez, J.J. — *Stochastic dynamics of neural networks*. — IEEE Trans. SMC-16, 73–83 (1986).
- Peretto, P., Niez, J.J. — *Long term memory storage capacity of multi-connected neural networks*. — Biol. Cybern. 54, 53–63 (1986).
- Peretto, P., Niez, J.J. — *Collective properties of neural networks*. — In ‘Disordered systems and biological organization’, Les Houches Session (1985), Bienenstock, E. et al. Eds., NATO ASI, series F20, pp. 171–185 (1986).
- Peretto, P. — *On models of short and long term memories*. — In Proceedings Conf. ‘Computer Simulation in Brain Science’, Cottrill, R. Ed., Cambridge University Press, Cambridge, pp. 88–103 (1988).
- Peretto, P. — *On learning rules and memory storage abilities of asymmetrical neural networks*. — J. Phys., Paris 49, 711–726 (1988).
- Peretto, P. — *Discrete linear programming problems solved by digital neural networks*. — In ‘Journées d’Électronique’, Neirynck, J. Ed., Presses Polytechniques Romandes, Lausanne, pp. 117–130 (1989).
- Peretto, P. — *Learning learning sets in neural networks*. — Int. J. Neural Syst., (World Scientific, Singapore) 1, 31–40 (1989).
- Peretto, P. — *On the dynamics of memorization processes*. — Neural Networks, (Pergamon Press) N.Y., 1, 309–322 (1988).
- Peretto, P. — *The semi-parallel architectures of neuro-computers*. — In Proc. NATO-ARW Conf.: ‘Neurocomputing, Algorithms, Architectures, and Applications’, Fogelman, F. Ed., Springer, Berlin (1990).
- Perkel, D. — *Logical neurons: The enigmatic legacy of Warren McCulloch*. — Trends Neurosci. 11, 9–12 (1988).
- Perrett, D.I., Rolls, E.T., Caan, W. — *Visual neurones responsive to faces in the monkey temporal cortex*. — Exp. Brain Res. 47, 329–342 (1982).
- Perron, O. — Math. Ann. 64, 248 (1907).
- Personnaz, L., Guyon, I., Dreyfus, G. — *Collective computational properties of neural networks: New learning mechanisms*. — Phys. Rev. A 34, 4217–4227 (1986).
- Personnaz, L., Guyon, I., Dreyfus, G., Toulouse, G. — *A biologically constrained learning mechanism in networks of formal neurons*. — J. Stat. Phys. 43, 411 (1986).
- Personnaz, L., Guyon, I., Dreyfus, G. — *Information storage and retrieval in spin-like neural networks*. — J. Physique Lett. 46, L359–L365 (1985).
- Personnaz, L., Guyon, I., Dreyfus, G. — *High order neural networks: Information storage without errors*. — Europhys. Lett. 4, 863–867 (1987).
- Peterson, C., Söderberg B.. — *A new method for mapping optimization problems onto neural networks*. — Int. J. Neural Syst. 1, 3–22 (1989).
- Peterson, C. — *Parallel distributed approaches to combinatorial optimization: Benchmark studies on TSP*. — Preprint, University of Lund, Sweden, N° LU TP 90-2 (1990).
- Pilyshyn, Z.W. — *Computation and cognition: Issues in the foundations of cognitive science*. — Behav. Brain Sci. 3, 111–169 (1980).

- Pineda, F. — Generalization of back-propagation to recurrent neural networks. — Phys. Rev. Lett. 59, 2229–2232 (1987).
- Pitts, W., McCulloch, W.S. — How we know universals: The perception of auditory and visual forms. — Bull. Math. Biophys. 9, 127 (1947).
- Poggio, T., Koch, C. — An analog model of computation for the ill-posed problems of early vision. — MIT Artif. Intell. Lab., AI Memo 783 (1984).
- Poggio, T., Torre, V., Koch, C. — Computational vision and regularization theory. — Nature 317, 314–319 (1985).
- Poggio, T., Edelman, S. — A network that learns to recognize three-dimensional objects. — Nature 343, 263–266 (1990).
- Pöppel, G., Krey, U. — Dynamical learning processes for recognition of correlated patterns in symmetric spin glass models. — Europhys. Lett. 4, 979–985 (1987).
- Prager, R., Harrison, T., Fallside, F. — Boltzmann machines for speech recognition. — Comput. Speech Lang. 1, 3–27 (1987).
- Pribram, K.H. — The role of analogy in transcending limits in the brain sciences. — Daedalus, Proc. Am. Acad. Arts Sci. 109, 19–38 (1980).
- Provost, J.P., Vallee, G. — Ergodicity of the coupling constants and the symmetric n -replicas trick for a class of mean-field spin-glass models. — Phys. Rev. Lett. 50, 598–600 (1983).
- Psaltis, D., Fahrat, N. — Optical information processing based on an associative-memory model of neural nets with thresholding and feedback. — Opt. Lett. 10, 98–100 (1985).
- Psaltis, D., Park, C., Hong, J. — Higher order associative memories and their optical implementations. — Neural Networks 1, 149–163 (1988).
- Psaltis, D., Lin, S., Yamamura, A., Gu, X.G., Hsu, K., Brady, D.. — Optoelectronic implementations of neural networks. — IEEE Commun Mag. 37–40 (November 1989).
- Qian, N., Sejnowski, T. — Predicting the secondary structure of globular proteins by using neural network models. — J. Mol. Biol. 202, 865–884 (1988).
- Quinlan, J.R. — Induction of decision trees. — Mach. Learn. 1, 81–106 (1986).
- Quinton, P. — Les hyper-ordinateurs. — La Recherche 167, 740–749 (1985).
- Rammal, R., Angles d'Auriac, Douçot, B. — On the degree of ultrametricity. — J. Physique. Lett. 46, L945–L952 (1985).
- Rammal, R., Toulouse, G., Virasoro, M. — Ultrametricity for physicists. — Rev. Mod. Phys. 58, 765–788 (1986).
- Rapoport, A. — Applications of game-theoretic concepts in biology. — Bull. Math. Biol. 47, 161–192 (1985).
- Rauschecker, J.P., Singer W. — Changes in the circuitry of the kitten visual cortex are gated by postsynaptic activity. — Nature 280, 58–60 (1979).
- Rauschecker, J.P., Singer, W. — The effects of early visual experience on cat's visual cortex and their possible explanation by Hebb synapses. — J. Physiol. (London) 310, 215–239 (1981).

- Reichardt, W., Guo, A. — *Elementary pattern discrimination (Behavioral experiments with the fly Musca domestica)*. — Biol. Cybern. 53, 285–306 (1986).
- Reichardt, W., Poggio, T. — *Figure-ground discrimination by relative movement in the visual system of the fly. I: Experimental results*. — Biol. Cybern. 35, 81–100 (1979). — *II: Towards the neural circuitry (with Hausen, K.)*. — Biol. Cybern. Suppl. 46, 390–394 (1983).
- Reid, R., Frame, S. — *Convergence in iteratively formed correlation matrix memories*. — IEEE Trans. C-24, 827–830 (1975).
- Reilly, D., Cooper, L., Elbaum, C. — *A neural model for category learning*. — Biol. Cybern. 45, 35–41 (1982).
- Rescorla, R.A., Wagner, A.R. — *A theory of Pavlovian conditioning: Variation in the effectiveness of reinforcement and non-reinforcement*. — In 'Classical Conditioning, II: Current Research and Theory', Black, H., Prokasy, W. Eds., Appleton-Century-Crofts, N.Y. (1972).
- Riehle, A., Franceschini, N. — *Motion detection in flies: Parametric control over ON-OFF pathways*. — Exp. Brain Res. 54, 390–394 (1984).
- Rolls, E.T. — *The representation and storage of information in neuronal networks in the primate cerebral cortex and hippocampus*. — In 'The Computing Neuron', Durbin, R., Miall, C., Michinson, G. Eds., pp. 125–159, Addison-Wesley, Wokingham (1989).
- Ronacher, B. — *Human pattern recognition: Evidence for a switching between strategies in analyzing complex stimuli*. — Biol. Cybern. 51, 205–210 (1984).
- Roney, K.J., Shaw, G.L. — *Analytic study of assemblies of neurons in memory storage*. — Math. Biosci. 51, 25–41 (1980).
- Roney, K.J., Scheibel, A.B., Shaw, G.L. — *Dendritic bundles: Survey of anatomical experiments and physiological theories*. — Brain Res. Rev. 1, 225–271 (1979).
- Rosenblatt, F. — *The perceptron, a probabilistic model for information storage and organization in the brain*. — Psych. Rev. 62, 386–407 (1958).
- Rosenfeld, R., Touretzky, D. — *Four capacity models for coarse-coded symbol memoires*. — Carnegie Mellon University Report CMU-CS-87-182 (1987).
- Rothman, D.H. — *Large near-surface anomalies, seismic reflection data, and simulated annealing*. — Thesis, Stanford University (1985).
- Rubel, L. — *The brain as an analog computer*. — J. Theor. Neurobiol. 4, 73–81 (1985).
- Rubin, N., Sompolinsky, H. — *Neural networks with low local firing rates*. — Europhys. Lett. 10, 465–470 (1989).
- Rujàn, P. — *Searching for optimal configurations by simulated tunneling*. — Z. Phys. B 73, 391–416 (1988).
- Rujàn, P. — *Learning and architectures of neural networks*. — In 'Models of Brain Function', Cottrill, R. Ed., Cambridge University Press, Cambridge (1989).
- Rujàn, P. — *A geometric approach to learning in neural networks*. — In Proc. of IJCNN (Mag. Joint Conf. Neural Networks) held in Washington, DC, pp. II-105–110 (1989).

- Rujàn, P., Evertsz, C., Lyklema, J. — *A Laplacian walk for the travelling salesman.* — *Europhys. Lett.* 7, 191–195 (1988).
- Rujàn, P., Marchand, M. — *Learning by minimizing resources in neural networks.* — Preprint, IFK, Jülich (1989).
- Rumelhart, D., Hinton, G., Williams, R. — *Learning internal representations by error propagation.* — In 'Parallel distributed processing. Explorations in the microstructure of cognition', vol 1, MIT Press, Cambridge MA (1986).
- Rumelhart, D., Hinton, G., Williams, R. — *Learning representations by back-propagating errors.* — *Nature* 323, 533–536 (1986).
- Saarinen, J., Kohonen, T. — *Self-organized formation of colour maps in a model cortex.* — *Perception* 14, 711–719 (1985).
- Sakitt, B., Barlow, H. — *A model for the economical encoding of the visual image in the cerebral cortex.* — *Biol. Cybern.* 43, 97–108 (1982).
- Salu, Y. — *Learning and coding of concepts in neural networks.* — *Biosyst.* 18, 93–103 (1985).
- Salu, Y. — *Theoretical models and computer simulations of neural learning systems.* — *J. Theor. Biol.* 111, 31–46 (1984).
- Schrager, J., Hogg, T., Huberman, B. — *A graph-dynamic model of the power law of practice and the problem-solving fan-effect.* — *Science* 242, 414–416 (1988).
- Seitz, C. — *Engineering limits on computer performance.* — *Phys. Tod.*, 38–45 (May 1984).
- Seitz, C. — *The Cosmic Cube.* — *Commns ACM.* 28, 22–33 (1985).
- Sejnowski, T. — *Neural populations revealed.* — *Nature* 332, 306 (1988).
- Sejnowski, T., Rosenberg, C.R. — *Parallel networks that learn to pronounce English text.* — *Complex Syst.* 1, 145–168 (1987).
- Sejnowski, T., Koch, C., Churchland, P. — *Comp. Neurosci.* — *Science* 241, 1299–1306 (1988).
- Silverston, A.I., Moulins, M. — *Are central pattern generators understandable?* — *Behav. Brain Sci.* 3, 535–571 (1980).
- Shakelford, B. — *Neural data structures: Programming with neurons.* — *Hewlett-Packard J.*, 69–78 (June 1989).
- Shannon, C. — *A mathematical theory of communication.* — *Bell Syst. Tech. J.* 27, 379–423 (1948).
- Shaw, G.L., Harth, E., Scheibel, A. — *Cooperativity in brain function: Assemblies of approximately 30 neurons.* — *Exper. Neurol.* 77, 324–358 (1982).
- Shaw, G.L., Roney, K.J. — *Analytic solution of a neural network theory based on an Ising spin system analogy.* — *Phys. Lett.* 74 A, 176–180 (1979).
- Shaw, G.L., Silverman, D.J., Pearson, J.C. — *Trion model of cortical organization: Toward a theory of information processing and memory.* — In 'Brain Theory', Palm G. and Aertsen A. Eds., Springer, Berlin (1986).
- Shaw, G.L., Silverman, D.J., Pearson, J.C. — *Model of cortical organization embodying a basis for a theory of information processing and memory recall.* — *Proc. Natl Acad. Sci. USA* 82, 2364–2368 (1985).

- Shepard, R.N., Metzler, J. — *Mental rotation of three-dimensional objects.* — Science 171, 701–703 (1971). Also Lynn Cooper and Roger Shepard. — *Le retournement mental des objets.* — Pour la Science, 40–47 (1985).
- Sherlock, R. — *Analysis of the behavior of Kauffman binary networks. I. State space description and the distribution of limit cycle lengths.* — Bull. Math. Biol. 41, 687–705 (1979). — *II. The state cycle fraction for networks of different connectivities.* — Bull. Math. Biol. 41, 707–724 (1979).
- Shiino, M., Nishimori, H., Ono, M. — *Nonlinear master equation approach to asymmetrical neural networks of the Hopfield-Hemmen type.* — J. Phys. Soc. Japan 58, 763–766 (1989).
- Shinomoto, S. — *Memory-maintenance in neural networks.* — J. Phys. A: Math. Gen. 20, L1305–1309 (1987).
- Shiozaki, A. — *Recollection ability of three-dimensional correlation matrix associative memory.* — Biol. Cybern. 50, 337–342 (1984).
- Singer, W. — *Hebbian modification of synapses.* — In ‘Synaptic Modification, Neuron Selectivity, and Nervous System Organization’, Levy, W., Anderson, J., Lehmkhle, S. Eds., Lawrence Erlbaum, Hillsdale, NJ , p. 35 (1985).
- Sirat, J.A., Jorand, D. — *Third order Hopfield networks: Extensive calculations and simulations.* — Preprint, LEP, Philips Laboratory, Limeil, France (1988).
- Skilling, J. — *The maximum entropy method.* — Nature. 309, 748–749 (1984).
- Sklansky, J. — *On the Hough technique for curve detection.* — IEEE Trans. C-27, 923–926 (1978).
- Sompolinsky, H. — *Neural networks with nonlinear synapses and a static noise.* — Phys. Rev. A 34, 2571–2574 (1986).
- Sompolinsky, H., Kanter, I. — *Temporal association in asymmetric neural networks.* — Phys. Rev. Lett. 57, 2861–2864 (1986).
- Sompolinsky, H., Crisanti, A., Sommers, H. — *Chaos in random neural networks.* — Phys. Rev. Lett. 61, 259–262 (1988).
- Sompolinsky, H. — *Statistical mechanics of neural networks.* — Phys. Tod. (December 1988).
- Sourlas, N. — *Multilayer neural networks for hierarchical patterns.* — Preprint, École Normale Supérieure N° 88/22, Paris (1988).
- Spitzner, P., Kinzel, W. — *Freezing transition in asymmetric random neural networks with deterministic dynamics.* — Preprint, to appear in Z. Phys. B (1989).
- Sprecher, D.A. — *A survey of solved and unsolved problems on superpositions of functions.* — J. Approx. Theory 6, 123–134 (1972).
- Stanton, P., Sejnowski, T. — *Associative long-term depression in the hippocampus: Evidence for anti-Hebbian synaptic plasticity.* — Preprint, John Hopkins University, Baltimore, MD (1988).
- Stein., R. — *Parallel processing: T800 and counting.* — BYTE, 287–296 (November 1988).
- Stein, R.B., Leung, K.V., Mangeron, D., Oguztoreli, M.N. — *Improved neuronal models for studying neural networks.* — Kybernetik 15, 1–9 (1974).

- Stein, R.B., Leung, K.V., Oguztöreli, M.N., Williams, D.W. — *Properties of small neural networks.* — Kybernetik 14, 223–230 (1974).
- Stein, J.F. — *The role of the cerebellum in the visual guidance of movement.* — Nature 325, 217–221 (1986).
- Stent, G.S. — *A physiological mechanism for Hebb's postulate of learning.* — Proc. Natl Acad. Sci. 70, 997–1001 (1973).
- Storm-Mathisen, J. — *Localization of putative transmitters in the hippocampal formation.* — In 'Functions of the Septo-hippocampal System', Ciba foundation, Elsevier (1978).
- Stornetta, S., Hogg, T., Huberman, B. — *A dynamical approach to temporal pattern processing.* — In Proc. IEEE Conf. 'Neural Information Processing Systems', Denver CO (1987).
- Sutherland, S. — *Parallel distributed processing.* — Nature 323, 486 (1986).
- Sutton, R., Barto, A. — *Toward a modern theory of adaptive networks: Expectation and prediction.* — Psychol. Rev. 88, 135–170 (1981).
- Suydam, W. — *AI becomes the soul of the new machines.* — Computer Design, 55–70 (February 1986).
- Swanson, L.W. — *The anatomic organization of septo-hippocampal projection.* — In 'Functions of the Septo-hippocampal System', Ciba foundation, Elsevier (1978).
- Swanson, L. — *Neuropeptides. New vistas on synaptic transmission.* — Trends Neurosci., 294–295 (August 1983).
- Swenden, R., Wang, J.S. — *Replica Monte Carlo simulation of spin-glasses.* — Phys. Rev. Lett. 57, 2607–2609 (1986).
- Swindale, N.V. — *A model for the formation of orientation columns.* — Proc. Roy. Soc. (London) B 215, 211–230 (1982).
- Szentagothai, J. — *The 'module concept' in cerebral cortex architecture.* — Brain Research 95, 475–496 (1975).
- Szentagothai, J. — *The modular architectonic principle of neural centers.* — Rev. Physiol. Biochem. Pharmacol. 98, 11–61 (1983).
- Tank, D., Hopfield, J.J. — *Simple 'neural' optimization networks: An A/D converter, signal decision circuit and a linear programming circuit.* — IEEE Trans. CAS 33, 533–541 (1986).
- Tank, D., Hopfield, J.J. — *Neural computation by time compression.* — Proc. Natl Acad. Sci. USA 84, 1896–1900 (1987).
- Tank, D.W., Hopfield, J.J. — *Simple optimization networks: An A/D converter and a linear programming circuit.* — IEEE Trans. CAS-33, 533–541 (1986).
- Tarjan, R.E. — *Complexity of combinatorial algorithms.* — SIAM Rev., 457–491 (July 1978).
- Tesauro, G. — *Simple neural models of classical conditioning.* — Biol. Cybern. 55, 187–200 (1986).
- Tesauro, G. — *Scaling relationships in back-propagation learning: Dependence on training set size.* — Complex Syst. 1, 367–372 (1987).

- Tesauro, G., Sejnowski, T. — *A parallel network that learns to play backgammon.* — Art. Intell. 39, 357–390 (1989).
- Thakoor, A.P., Moopenn, A., Lambe, J., Khanna, S.K. — *Electronic hardware implementations of neural networks.* — Appl. Opt. 26, 5085–5092 (1987).
- Thomas, R. — *Logical description, analysis, and synthesis of biological and other networks comprising feedback loops.* — In 'Aspects of Chemical Evolution', Nicolis, G. Ed., Wiley, N.Y., pp. 247–282 (1980).
- Thompson, R.S., Gibson, W.G. — *Neural Model with Probabilistic Firing Behavior. 1: General Considerations. 2: One- and Two-Neuron Networks.* — Math. Biosci. 56, 239–285 (1981).
- Thompson, R.S. — *A model for basic pattern generating mechanisms in the lobster stomatogastric ganglion.* — Biol. Cybern. 43, 71–78 (1982).
- Tiberghien, G., Cauzinille, E., Mathieu, J. — *Pre-decision and conditional research in long-term recognition memory.* — Acta Psychol. 43, 329–346 (1979).
- Toffoli, T. — *CAM: A high-performance cellular-automaton machine.* — Physica 10D, 195–204 (1984).
- Toulouse, G., Dehaene, S., Changeux, J.P. — *A spin-glass model of learning by selection.* — Proc. Natl Acad. Sci. USA 83, 1695–1698 (1986).
- Touretzky, D., Pomerleau, D. — *What is hidden in the hidden layers?* — BYTE, 227–233 (August 1989).
- Treisman, A. — *A feature-integration theory of attention.* — Cogn. Psychol. 12, 97–136 (1980).
- Treisman, A. — *Properties, parts and objects.* — In 'Handbook of Perception and Performance', vol. 2, Boff, K., Kaufman, L., Thomas, J., Eds., Wiley, N.Y. (1986).
- Treisman, A. — *L'identification des objets visuels.* — Pour la Science, pp. 50–60 (Jan. 1987).
- Treves, A., Amit, D. — *Low firing rates: An effective Hamiltonian for excitatory neurons.* — J. Phys. A: Math. Gen. 22, 2205–2226 (1989).
- Trugman, S.A. — *General theory of inhomogeneous systems, based on maximum entropy.* — Phys. Rev. Lett. 57, 607–610 (1986).
- Tsodyks, M.V. — *Associative memory in asymmetric diluted network with low level of activity.* — Europhys. Lett. 7, 203–208 (1988).
- Tsodyks, M.V., Feigel'man, M.V. — *The enhanced storage capacity in neural networks with low activity level.* — Europhys. Lett. 6, 101–105 (1988).
- Tsuda, I., Koerner, E., Shimizu, H. — *Memory Dynamics in Asynchronous Neural Networks.* — Prog. Theor. Phys. 78, 51–71 (1987).
- Tsukada, M., Aihara, T., Hauske, G. — *Redundancy reducing processes in single neurons.* — Biol. Cybern. 50, 157–165 (1984).
- Tsutsumi, K., Matsumoto, H. — *A synaptic modification algorithm in consideration of the generation of rhythmic oscillation in a ring neural network.* — Biol. Cybern. 50, 419–430 (1984).
- Vallet, F. — *The Hebb rule for learning linearly separable Boolean functions: Learning and generalization.* — Europhys. Lett. 8, 747–751 (1989).

- Valiant, L.G. — *A theory of the learnable.* — ACM Communications 27, 1134–1142 (1984).
- Van Essen, D. — *Visual areas of the mammalian cerebral cortex.* — Ann. Rev. Neurosci. 2, 227–263 (1979).
- Van Essen, D., Maunsell, J. — *Hierarchical organization and the functional streams in the visual cortex.* — Trends Neurosci., Elsevier, 370–375 (September 1983).
- Van Hemmen, J.L. — *Classical spin-glass model.* — Phys. Rev. Lett. 49, 409–412 (1982).
- Van Hemmen, J.L., Van Enter, A.C. — *Chopper model of pattern recognition.* — Phys. Rev. A 34, 2509–2512 (1985).
- Vedenov, A.A., Levchenko, E.B. — *On a class of non-linear systems applied to memory.* — JETP 41, 328–331 (1985) (in Russian).
- Venkatesh, S. — *Epsilon capacity of neural networks.* — In ‘Neural Networks for Computing’, Denker, J.S. Ed., AIP Conf. vol. 151, pp. 440–464 (1986).
- Venkatesh, S., Psaltis, D. — *Linear and logarithmic capacities in associative neural networks.* — IEEE Trans. IT-35, 558–568 (1989).
- Ventriglia, F. — *Kinetic theory of neural systems: Analysis of the activity of the two-dimensional model.* — Biol. Cybern. 46, 93–99 (1983).
- Vichniac, G. — *Simulating physics with cellular automata.* — Physica 10 D, 96–116 (1984).
- Victor, J. — *Bell Labs models parallel processor on neural networks.* — Mini-Micro Syst., 43–51 (August 1986).
- Virasoro, M. — *The effect of synapses destruction on categorization by neural networks.* — Europhys. Lett. 4, 293–298 (1988).
- Virasoro, M. — *Analysis of the effects of lesions on a perceptron.* — J. Phys. A: Math. Gen. 22, 2227–2232 (1989).
- Von der Malsburg, C. — *Self-organization of orientation sensitive cells in the striate cortex.* — Kybernetik 14, 85–100 (1973).
- Von der Malsburg, C. — *How are nervous structures organized?* — in Proc. Mag. Symp. Synergetics: ‘Synergetics of the Brain’, Basar, E., Flohr, H., Haken, H., Mandell, A.J. Eds., Springer, Berlin., pp. 238–249 (1983).
- Von der Malsburg, C., Cowan, J. — *Outline of a theory for the ontogenesis of iso-orientation domains in visual cortex.* — Biol. Cybern. 45, 49–56 (1982).
- Von der Malsburg, C., Bienenstock, E. — *A neural network for the retrieval of superimposed connection patterns.* — Europhys. Lett. 3, 1243–1249 (1987).
- Von der Malsburg, C., Schneider, W. — *A neural cocktail-party processor.* — Biol. Cybern. 54, 29–40 (1986).
- Von der Malsburg, C., Willshaw, D.J. — *Differential equations for the development of topological nerve fibre projections.* — SIAM-AMS Proc. 13, 39–47 (1981).
- Von Neumann, J. — *Probabilistic logics and the synthesis of reliable organisms from unreliable components.* — In ‘Cerebral Mechanisms of Behavior: The Hixon Symposium’, Jeffres, L.A. Ed., Wiley, N.Y. (1951).

Von Neumann, J. — *The general and logical theory of automata*. — In 'Cerebral Mechanisms of Behavior: The Hixon Symposium', Jeffres, L.A. Ed., pp. 1-32, Wiley, N.Y. (1951).

Waibel, A., Hampshire, J. — *Building blocks for speech*. — BYTE, 235-242 (August 1989).

Wain, C. — *Concurrent computing: A new age in supercomputer architecture*. — Solutions, Intel Corporation, 6-9 (May 1985).

Waldrop, M. — *Natural language understanding*. — Science 224, 372-373 (1984).

Wallace, D.J. — *Memory and learning in a class of neural network models*. — Preprint, Edinburgh 86/363 (1986).

Wallace, D.J. — *Spin-glass models of neural networks: size dependence of memory properties*. — In Proc. Workshop 'Advances in Lattice Gauge Theory', Tallahassee, World Scientific, Edinburgh (1985) and Plenum (1986).

Waltz, D.L., Pollack, J.B. — *Massively parallel parsing: A strongly interactive model of natural language interpretation*. — Cognitive Science 9, 51-74 (1985).

Weisbuch, G. — *Un modèle de l'évolution des espèces à trois niveaux, basé sur les propriétés globales des réseaux booléens*. — C. R. Acad. Sci. Paris 298, 375-378 (1984).

Weisbuch, G., Fogelman-Soulie, F. — *Scaling laws for the attractors of Hopfield networks*. — J. Physique Lett. 46, 623-630 (1985).

Weisbuch, G., Stauffer, D. — *Phase transition in cellular random Boolean nets*. — J. Phys., Paris 48, 11-18 (1987).

Weiss, V. — *Memory as a macroscopic ordered state by entrainment and resonance in energy pathways*. — Psychogenetik der Intelligenz (GRKG/ Human Kybernetik, Borgmann K.G., Dortmund) 27, 201-221 (1986).

Widrow, B., Hoff, M.E. — *Adaptive switching circuits*. — 1960 IRE (Institute of Radio Engineers) WESCON (Western Electronic Show and Convention) Conv. Record, Part 4, 96-104 (August 1960).

Widrow, B. — *Generalization and information storage in networks of Adaline 'neurons'*. — In 'Self-Organizing Systems', Yovits *et al.* Eds., Spartan, Washington DC, pp. 435-461 (1962).

Widrow, B., Angell, J.B. — *Reliable, trainable networks for computing and control*. — Aerospace Eng. 21, 78-123 (1962).

Widrow, B., Gupta, N.K., Maitra, S. — *Punish/Reward: Learning with a critic in adaptive threshold systems*. — IEEE Trans. SMC-3, 455-465 (1973).

Wiesel, T., Hubel, D. — *Comparison of the effects of unilateral and bilateral eye closure on cortical unit response in kittens*. — J. Neurophysiol. 28, 1029-1040 (1963).

Williams, T. — *Optics and neural nets: Trying to model the human brain*. — Computer Design. 47-62 (March 1987).

Willshaw, D.J., Buneman, O.P., Longuet-Higgins, H.C. — *Non-holographic associative memory*. — Nature 222, 960-962 (1969).

- Willshaw, D., Von der Malsburg, C. — *How patterned neural connections can be set up by self-organization.* — Proc. Roy. Soc., London B 194, 431–445 (1976).
- Willshaw, D.J., Von der Malsburg, C. — *A marker induction mechanism for the establishment of ordered neural mappings: Its application to the retinotectal problem.* — Phil. Trans. Roy. Soc. (London) B 287, 203 (1979).
- Willwacher, G. — *Do different synaptic coupling values represent simultaneous and sequenced firing of neurons?* — J. Theor. Biol. 2, 155–160 (1983).
- Willwacher, G. — *Fähigkeiten eines assoziativen Speichersystems im Vergleich zu Gehirnfunktionen.* — Biol. Cybern. 24, 181–198 (1976).
- Willwacher, G. — *Storage of a temporal pattern sequence in a network.* — Biol. Cybern. 43, 115–126 (1982).
- Wilson, A.C., Wilson, D.W. — *Neural networks: Applications-specific problem solvers.* — Elec. Syst. Design Mag. 30–32 (February 1989).
- Wilson, H.R., Cowan, J.D. — *A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue.* — Kybernetik 13, 55–80 (1973).
- Winder, R.O. — *Bounds on threshold gate realizability.* — IEEE Trans. EC-12, 561–564 (1963).
- Wolff, J.R., Wagner, G.P. — *Self-organization in synaptogenesis: Interaction between the formation of excitatory and inhibitory synapses.* — In 'Synergetics of the Brain', Basar, E., Flöhr, H., Haken, H., Mandell, A.J. Eds., Springer, Berlin, pp. 50–59 (1983).
- Wolfram, S. — *Statistical mechanics of cellular automata.* — Rev. Mod. Phys. 55, 601–644 (1983).
- Wolfram, S. — *Universality and complexity in cellular automata.* — Physica 10 D, 1–35 (1984).
- Wolfram, S. — *Cellular automata as models of complexity.* — Nature 311, 419–424 (1984).
- Wong, R., Harth, E. — *Stationary states and transients in neural populations.* — J. Theor. Biol. 40, 77–106 (1973).
- Wong, K., Sherrington, D. — *Storage properties of randomly connected Boolean neural networks for associative memory.* — Europhys. Lett. 7, 197–202 (1988).
- Wong, K., Sherrington, D. — *Theory of associative memory in randomly connected Boolean neural networks.* — J. Phys. A: Math. Gen. 22, 2233–2263 (1989).
- Wooldridge, D. — *Memory neuron: Synapse microchemistry for the memory component of a neuroconnective brain model.* — Proc. Natl Acad. Sci. USA 77, 3019–3023 (1980).
- Wright, J., Kydd, R., Lees, G. — *State-changes in the brain viewed as linear steady-states and non-linear transitions between steady-states.* — Biol. Cybern. 53, 11–17 (1985).
- Yasunaga, M., Masuda, N., Asai, M., Yamada, M., Masaki, A., Hirai, Y. — *A wafer scale integration neural network utilizing completely digital circuits.*

— In Proc. Mag. Joint Conf. Neural Networks (IJCNN, San Diego 89) IEEE, Washington (1989).

Yuhas, B., Goldstein, M., Sejnowski, T. — *Integration of acoustic and visual speech signals using neural networks.* — IEEE Commns Mag. 65-71 (November 1989).

Zak, M. — *Terminal attractors for addressable memory in neural networks.* — Phys. Lett. A 133, 218-222 (1988).

Zak, M. — *The least constraint principle for learning in neurodynamics.* — Phys. Lett. A 135, 25-28 (1989).

Zeki, S. — *The representation of colours in the cerebral cortex.* — Nature 284, 412-418 (1980).

Zipser, D., Andersen, R.A. — *A back-propagation programmed network that simulates responses properties of a subset of posterior parietal neurons.* — Nature 331, 679-684 (1988).

INDEX

- Action potential 29, 31, 32, 35, 39, 404
Adaline 10, 232, 233, 389
AGS parameter (Amit Gutfreund Sompolinsky) 115, 135
— order parameter 143, 149
Aleksander model 296
Analog
— devices 401
— neurocomputers 385
Annealing
— procedure 295
— protocol 328
Anti-Hebbian rule 52
Anti-learning 161
— processes 141
— rule 370
Antisymmetrical 167
Antisymmetrically
— connected 77
— connected networks 75
Aplysia 52, 94
Assignment problem 334, 335
Association 100, 178, 228, 367
— process 210, 416
Associative
— cortical areas 11
— learning rules 287, 232
— memory 11, 391, 400, 401
Asymmetrical 219, 342
— architecture 343
— hebbian rules 139
— interactions 240
— connected networks 78
Asymmetric connectivity matrix 259, 260
Asynchronous 72, 76, 98, 331
— dynamics 76, 77, 155, 340
— neurocomputer 396
Attention 55, 324, 325, 369, 416
Attractor 100, 170, 406
Auditory cortex 405
Average
— frequency 404
— over realizations 81, 110, 116, 118, 130
— over possible realizations 199
Back-propagation 282, 413
— algorithm 10, 269, 273, 276, 278, 373
Band 22, 23, 302
Bar orientation 51, 310
Basin 244, 380
— of attraction 67, 100, 104, 152, 171, 197, 244, 220, 221, 225, 248, 250, 251, 266, 292, 324, 370, 376
Behaviorism 2, 407
Bias 190, 192, 197, 243–245
Biased perceptron rule 243
Bipartition 333
Boltzmann
— machine 10, 286, 287, 323
— machine algorithm 288, 291, 292
— machine theorem 289
— statistics 114
— function 175
Boolean
— function 176, 179, 181, 183–186, 281, 286, 296
— mapping 174–176, 182, 184, 187, 281
— network 292, 293, 296, 415
Bursting 54
— activity 412
Caianiello equations 59, 61
Catastrophe overcrowding 246
Categorization 235, 236, 282
— problems 279
— tasks 10
Central nervous systems 14

- Central pattern generators 325
- Cerebellum 17, 26, 325, 366–368
- Chaotic 68
 - behavior 66
- Chen algorithm 347
- CHIR 278, 279
 - algorithm 279, 280
- Classical conditioning 12, 42, 43, 45, 47, 52, 100, 168, 170, 174
- Cocktail party problem 316
- Code 319, 404
- Collective
 - behavior 8
 - phenomena 376
- Columnar activities 74
- Columns 16, 23, 25, 28, 74, 254, 414
- Combinatorial optimization 325, 326, 331, 400
 - problem 329
- Complexity 181
- Conditioning 5, 99
- Connected network 376
- Convergent architecture 383
- Correlated 264
 - patterns 248, 251, 264
- Cortex 16, 18, 21–23, 25, 55, 254, 292, 299, 309, 310, 324, 358, 368, 369, 375
- Cortical
 - area 18, 23, 25
 - bands 21, 28, 302
 - columns 402
 - layer 303
 - maps 21, 22, 311
 - microcolumn 16
- Cost 337, 343, 344, 349, 363
 - energy 328, 340
 - function 230–232, 234, 243, 244, 270, 274, 277, 280, 301, 303, 328, 337, 338
- Cover
 - limit 187, 196, 197
 - theorem 239
- Credit assignment 269
 - problem 269
- Critical slowing down 128
- Cyclic attractor 67, 76, 78, 153, ...
- Cyclic attractor ... 155, 297
- Darwinian
 - competition mechanism 318
 - learning 318
 - selection 166
- Delay 34, 59, 66, 70, 74, 159, 405, 316, 406
- Detailed balance
 - equations 295
 - principle 86, 87, 330
 - probability 331
- Deterministic 112
 - dynamics 64, 67, 78
- Devil staircase 64
- Discrimination 48
- Distribution of delays 71
- Divergent architecture 383
- Dream sleep 55, 370
- E.A. parameter (Edwards-Anderson) 115, 133, 134, 135, 138, 143, 199, 200
- Elastic algorithm 345
- Energy 77, 78, 100, 101, 106, 112, 116, 118, 147, 197, 216, 295, 326, 328, 331, 332, 335, 340, 342, 344, 345, 354, 355–357, 362
 - function 76, 86, 88, 101, 106, 314, 230, 293
 - landscape 112, 379
- Ensemble 58, 69–71, 81, 91, 114
 - average 68, 131, 135
 - of systems 81
- Ergodic 59
- Error
 - process 12, 210, 232, 235, 269
 - function 230
- Expert system 12, 326, 372, 373
- Faithfulness 282, 284, 286, 289
- Faithful internal representation 283, 285, 287, 289
- Feature 21, 22
 - detectors 10, 21, 22, 310, 312, 315
- Feedforward 11, 269, 271, 301, 302, 312, 321, 351

- Feedforward
 - architecture 281
 - boolean networks 293
 - connections 11
 - layered 299
 - network 279, 270, 323
 - neural nets 9
- Feldman and Cowan dynamical equations 59
- Firing rate 32, 166
- Fixed point 67, 76, 78, 99, 100, 101, 112, 153, 174, 220, 230, 246, 343, 376
- Formal neuron 6
- Free energy 114, 115, 122, 130, 141, 144, 146, 199, 330
- Fully connected networks 299, 323, 358, 406
- Gate XOR 7
- Gaussian transform 115, 116, 142, 145, 203, 205
- Geman algorithm 354
- General-purpose neurocomputers 401
- Generalization 12, 48, 286, 321–323, 373, 375, 417
 - parameter 295
- Gerschgorin theorem 84, 86
- Glauber dynamics 71, 72, 74, 76, 78, 80, 82–86, 88, 90–92, 96, 98, 112, 125, 126
- Gradient
 - algorithm 230, 271, 277
 - dynamics 233, 235, 243, 246, 275, 292
- Grandmother 181, 364, 376
 - cell 179, 302, 320, 416
 - neural network 180
- Hamming
 - distance 231, 232, 255, 273, 295, 320, 357, 361
 - net 360
- Hard problem 181, 331, 333, 350
- Hebbian 10
- Hebbian diluted networks 149
- Hebbian
 - efficacies 194
 - layered networks 151
 - learning rule 100, 101, 103, 209, 369
 - models 99, 115
 - neural networks 103, 112, ...
 - neural networks ... 118, 124
 - paradigm 52
 - rule 9, 46, 51, 52, 100, 102, 150, 170, 173, 213, 214, 216, 220, 224, 234, 248, 251, 287, 314, 410
 - type rule 274
- Hetero-associative 228
- Hetero-synaptic junction 40, 54, 162, 170, 367, 368
- Hidden
 - layer 10, 179, 180, 181, 273, 279, 285, 375
 - neurons 181, 239
 - unit 176, 179, 183, 228, 269, 270, 278, 287–289, 291, 292, 302, 323
- Hierarchical
 - architecture 414
 - organization 13, 14, 16
 - structures 254
- Hierarchically
 - correlated patterns 253
 - organized patterns 266
 - structured 15
 - structured searches 254
- Hierarchy 16, 111
- Hippocampus 17, 26, 52, 55, 324, 367, 369, 405
- Homo-synapses 54
- Hopfield
 - dynamics 93
 - model 9, 133, 134, 137, 212
 - rule 245
- Hough transform 352
- Ill-defined problem 12, 174
- Ill-posed problem 326, 348, 349, 359
- Information 262, 263, 264, 266, 299, 322, 361, 370, 403, 404
- Inhibitory neuron 410
- Instantaneous average activities 88

- Instantaneous frequency 32, 404
- Instrumental conditioning 47
 - experiments 55
- Internal representation 10, 179, 228, 278–280, 269, 270, 278, 283, 286
- Invariant recognition 361
- Kohonen algorithm 305, 307, 371
- Kolmogorov theorem 182
- Lateral inhibition 303, 309, 313, 350
- Layer 22, 25, 151, 270, 302, 309, 312, 317, 324, 353, 373
 - Layered
 - feedforward networks 406
 - neural networks 12
 - systems 278
 - Learning 9
 - dynamics 41
 - set 47, 300
 - Limit
 - distribution 86, 87
 - storage capacity 105
 - Linear
 - neural networks 299
 - programming 338
 - programming problem 329, 343
 - separability 183
 - Linearly separable boolean functions 184–187
 - Little
 - dynamics 72–74, 78, 81–85, 87, 88, 90, 96, 98, 152, 153
 - matrix 85
 - type 74
 - Local field distributions 130
 - Long-range spin glasses 133
 - Long-term memory 48, 50, 54, 55, 246, 248, 324, 368, 369, 413
 - Low activity 250, 406, 408
 - Lyapunov function 74–76, 230, 293
 - Map coloring problem 331
 - Maps 22, 318
 - Master equation 67, 69, 82, 88
 - Mattis states 120
 - Maximum memory storage 104, ...
 - Maximum memory storage ... 195, 196, 200, 240, 265, 267
 - Maxwell-Boltzmann distribution 87
 - Mean field
 - approach 128
 - approximation 93, 121, 123, 124, 127, 128
 - technique 134
 - theory 124
 - Membrane potential 29, 31, 35, 37, 57, 93, 227, 382
 - Memorized patterns 102, 103, 110, 112, 115, 117, 118, 120, 153, 197
 - Memory 48, 99, 100, 101, 325
 - storage capacities 103, 137, 173, 196, 197, 200, 209, 214, 216–218, 225, 227, 229, 230, 251, 252, 257, 259, 261, 262, 265, 297, 358, 401, 410
 - Metropolis algorithm 295, 358
 - Mexican hat 309, 351
 - Microcolumns 23, 24, 28, 254, 414
 - Minover algorithm 239, 245
 - Modifiable synapse 411
 - Modular organization 414
 - Momentum gradient algorithm 277
 - Monte-Carlo 72, 125
 - Neocognitron 10
 - Net-Talk 374
 - Neural code 404
 - Neurocomputers 379
 - Non-linearly separable functions 273
 - Obsession 292, 302, 324, 370
 - Old hebbian
 - dynamics 260
 - rules 102
 - On-off cells 311, 315, 351
 - Optical
 - mask 390
 - parallel neurocomputer 390
 - techniques 402
 - Optimization 326
 - problem 274, 326, 328, 331, 338, 350
 - process 12

- Order parameter 114–117, 122, 127, 146, 168, 199, 201, 411
Orientation columns 314
Oscillators 405
Overcrowding catastrophe 209, 218, 370
Overlap 101, 115, 116, 118, 152, 200, 251, 252, 361, 412
- Pacemaker 166, 167
Palimpsest 215, 217
Parallel neurocomputer 382, 386
Partition function 86, 114, 116, ...
Partition function ... 141, 197, 199
Pattern
— invariance 361
— matching 358, 359
— recognition 12, 68, 254, 325
Perceptron 7, 10, 176, 183, 184, 187, 197, 230, 233–235, 238, 239, 278, 301
— architecture 176, 232
— learning algorithm 10, 230, 233, 234, 235, 239, 273, 278, 280, 282–284, 286
— learning principle 234
— principle 324
— rule 236, 238–240, 278, 280, 283
— theorem 236, 240
Perron
— eigenvalue 84, 85
— eigenvector 83, 84
Persistent states 408
Phase space 65, 72, 91, 112, 173, 288, 328, 331, 339, 379
— of interactions 173
Plastic 165, 312
— efficacies 161
Plasticity 16, 160, 212, 413
Pocket algorithm 238
Population selection theories 318
Primacy effects 48, 214, 216
Principal component 300, 302
Probabilistic threshold automata 78, 99
Problem of credit assignment 176
Projection
— algorithm 10, 218, 221, 227, ...
Projection
— algorithm ... 228, 229, 233, 251
— gradient algorithm 277
— learning algorithm 230
— operator 221, 223, 229
Prosopagnosia 267
Pseudo-inverse 228, 229, 67
- Random coding 320
Realization 115, 116, 118, 119, 130, 131
Recency effect 48, 214, 216
Recursive
— networks 9, 302, 323
— neural network 326
Refractory period 29, 32, 34, 36, 39, 57, 58, 64, 70, 71, 92
Relative entropy 289
Replica 141, 218
— method 139
— symmetry 144, 203
— symmetry approximation 203
— symmetry breaking 147
— technique 130, 141, 200, 229
Rescola Wagner rule 47, 219
Retinotopy 21, 316, 317
— model 304
Reward and punishment 10, 12
Robotics 371
- Saddle point 144, 203, 206, 207
— method 121, 203
Sample average 142, 201
Searchlight hypothesis 369
Self-averaging 118
Self-correcting 162
Self-organization 10, 210, 324, 347
— algorithm 304, 306
— model 304
— system 407
Semantic
— memory 51
— net 375, 377, 416
Semi-parallel
— architecture 392, 401
— convergent 391
— divergent 391

- Semi-parallel
 - machine 359, 383, 392
 - neurocomputer 383, 391
 - analog machine 400
- Serial neurocomputer 383, 385
- Sherrington-Kirkpatrick model 132
- Shift register 392
- Short-term memory 48, 49, 55, 215, 246
- Signal processing 350
- Simulated annealing algorithm 295, 330
- Sparse coding 259, 260, 320, 376, 408, 411
- Spin
 - glass 136, 141, 147, 199, 334, 412
 - phase 147
- Spurious states 106, 109, 111, 112, 115, 118, 120, 261
- Stabilization parameter 187, 189, 195, 196, 198, 218, 220, 231, 232, 234, 243, 244, 271
- Statistical average 69
- Steady
 - distributions 83
 - state 407
- Steepest
 - descent 121, 128
 - descent approximation 117
 - descent method 124, 127
 - gradient approximation 144
- Storage capacity 189, 196, 207
- Supervised learning 210
- Suzuki algorithm 295
- Symmetrical 86, 87, 105, 112, 124, 159, 225, 252, 326, 327, 376
 - connections 291
 - interactions 87, 133
- Symmetrical networks 78
- Symmetrically 331
 - connected 75, 77, 78, 153
 - connected neural networks 87
 - Hebbian network 112
- Symmetry 230
- Synaptic
 - connections 269
 - efficacies 33, 35, 36, 38, 54, ...
- Synaptic
 - efficacies ... 94, 155, 174, 212, 215, 218, 221, 230, 231, 234, 257, 274, 280, 288, 289, 291, 300–302, 304–307, 314, 316–318, 341, 358, 367, 369, 370, 373, 381, 385, 396, 398, 401
 - plasticity 54
 - transmission 25, 33, 35, 57
- Synchronization 72, 73, 369, 405
- Synfire chain 316, 404–406
- Systolic computation 392
 - design 392
- Teachers and classes 336
- Temporal
 - coincidence 367
 - sequences 155, 165
 - sequences of patterns 406
- Ternary synapses 162, 166
- Thermal
 - annealing 291, 358
 - annealing algorithm 307
 - annealing procedure 278, 330, 350, 356
 - average 81, 88, 115, 130, 131, 232, 288, 291
- Thermometer coding 320
- Threshold plasticity 166
- Tiling algorithm 279, 282, 286, 323, 375
- Time
 - averaging 313, 364
 - warping 366
- Topological
 - mapping algorithm of Kohonen 306
 - maps 302
- Training set 295, 296
- Transition matrix 67, 72, 81–83, 92
- Transputer 386
- Traveling salesman problem (TSP) 344, 348, 350
- Trial and error 295
- Ultrametric space 255
- Unbiased 190

- Unsupervised learning 210
- Visual
 - cortex 407
 - maps 22
- Volume 199, 200, 296, 373
 - of available solutions 198
 - of phase space 322
 - of solutions 174, 196, 209, 243, 257, 264
- Wallace Gardner algorithm 245
- Wallace tree adder 398
- Widrow Hoff rule 47, 219
 - algorithm 218, 226
 - learning rule 218, 219
 - rule 227, 228, 233, 245
- Willshaw model 260
- Winner-takes-all 318–320, 361, 363, 365
- XOR 10, 183, 184
 - function 181, 182, 184, 273