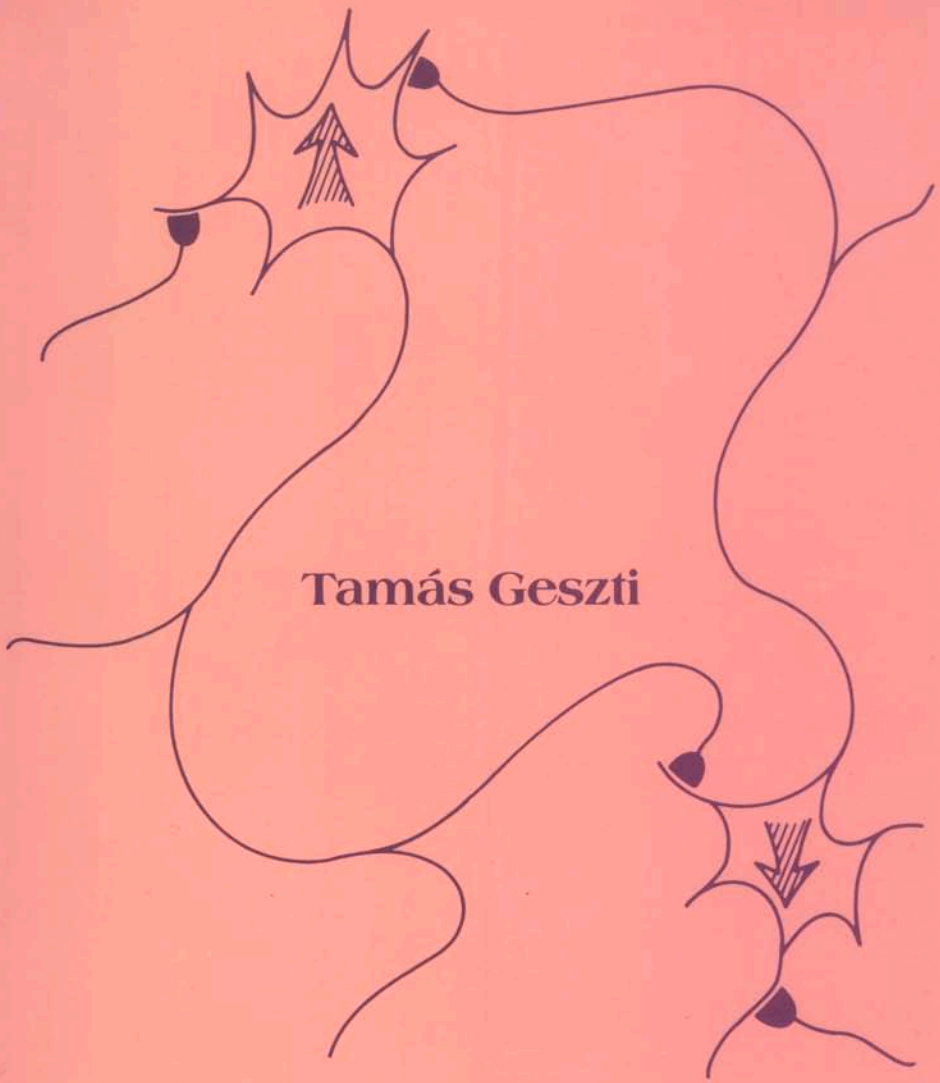
Physical Models of Neural Networks



Tamás Geszti

World Scientific

Physical Models of Neural Networks

This page is intentionally left blank

Physical Models of Neural Networks

Tamás Geszti
Department of Atomic Physics
Eötvös University
Budapest, Hungary

World Scientific
Singapore • New Jersey • London • Hong Kong

Published by

World Scientific Publishing Co. Pte. Ltd.,
P O Box 128, Farrer Road, Singapore 9128
USA office: 687 Hartwell Street, Teaneck, NJ 07666
UK office: 73 Lynton Mead, Totteridge, London N20 8DH

Library of Congress Cataloging-in-Publication Data

Geszti, Tamas.
Physical models of neural networks/by Tamas Geszti.
p. cm.
ISBN 9810200129

1. Neural circuitry — Models. 2. Neural computers. I. Title.
QP363.3.G47 1990
612.8'2'011--dc20 89-48133
CIP

Printed in Singapore by JBW Printers and Binders Pte. Ltd.

PREFACE

You all know what your heart is. It is a pump. It also looks like a pump, with its muscles capable to make it expand and contract and its valves letting the blood in and out when necessary. It is good mechanical engineering, well done for the purpose.

The same is true about your stomach: it is a chemical reactor, correctly devised and implemented for that.

But what would you say about your brain? You learn at school that it is the organ of thinking and regulation, but what should such an organ look like? Although many other metaphores have been mentioned in this context, probably the piece of engineering closest in function to a brain is a computer. However, nothing less similar in appearance than a human brain and a man-made computer; nothing like the straightforward connection between shape and function familiar from the examples of heart and stomach.

One of the less transparent metaphores says that in some respects – at least as far as associative memory storage and recall is concerned, but probably quite a bit beyond that – the brain is analogous to a spin-glass. This suggestion was advanced by the solid-state physicist John Hopfield in 1982, and marked the beginning of an important development in creating neural network models which are tractable by tools of theoretical physics, in particular, statistical physics. The results seem to have relevance to both brain research and the so-called neural computers, providing building principles for a class of the latter.

Neural network modelling is an enormous field of which physicists' models cover only a little but fascinating corner. Those interested in the lively development of the whole area should consult recent volumes of the journal "Biological Cybernetics" and the recently launched journal "Neural Networks".

The intention of this lecture notes volume is to offer the Reader an introduction to the subject from the physicist's viewpoint. Although the main subject is the Hopfield model and its descendants, feed-forward networks and their applications to computing tasks are also reviewed. The text is based on the author's

course given for physics and biophysics students at Eötvös University, Budapest, in 1987 and 1988.

It is a pleasure to thank here Jean-Pierre Nadal and John Hertz to whom I owe much of my insight to the subject, and my students Ferenc Pázmándi, Ferenc Czakó and István Csabai who helped me with many discussions and criticism on some parts of the manuscript. I thank Zsolt Frei for help in the final preparation of the manuscript. Whenever I felt uneasy about complications concerning spin-glasses, I had the comfort to make use of Imre Kondor's understanding of the subject (of course he is not responsible for cases when I ought to have felt uneasy but I did not). This book would not have been written without the inspiring atmosphere of the Physics Department of Limburgs Universitair Centrum in Diepenbeek (Belgium), with the kind hospitality of Professors Roger Serneels and Marc Bouten and their inexhaustible insistence on clear understanding. Most recently, Dr. Christian Van den Broeck of the same department and Professor Janusz Jędrzejewski visiting there from the University of Wrocław read the manuscript very carefully, detected a number of errors and asked questions suggesting improvements. Going back in time, when I started learning the subject, I profited from learning together and discussing with Róbert Németh. Last but not least, it cannot be left unmentioned that my interest in the subject was aroused by George Marx asking a couple of years ago, "What's all the excitement about brains and spin-glasses?"

T. Geszti

November 1989

CONTENTS

1 ASSOCIATIVE MEMORY AND ATTRACTORS

Have you ever been thinking about how enormous amount of computing it takes, to *see* something? There is no optical communication system in your brain: the images received in your eyes are coded by your nervous system in a fantastic richness of details, then processed and restored in your consciousness, correcting for changes of illumination, mood and many other circumstances.

Visual processing is very fast. However most mental processes take a manifest amount of time. To mention a task just a bit more involved than just vision, to recognize an old friend you haven't seen for ages may be a difficult decision problem about whether the one you see now is the same person you used to know before. For a little while you may feel pending between yes and no. Then you feel *attracted* by one of the two answers. The decision being taken, your thinking comes to a short rest.

This tendency of evolving towards a distinguished state carrying some significant meaning, usually a solution to some part of a task, which – if reached – is marked by a little rest, seems to be a very widespread scenario of mental processes on very different levels. Its dominating presence is reflected even in the traditional layout of opera music: scenes and recitatives visualize the dramatic evolution towards points of rest, which are then marked by arias. In the context of physical models of neural networks it has been clearly stated for the first time by Parisi (1986).

Simpler to model than a temporary halt is a single configuration or a limited set of them which is approached by the evolution of the system, and once reached, it is not left any more. Such a set of configurations is called an *attractor* of the dynamics. The simplest case is a single attracting configuration. That has a particular name: it is called an *attractive fixed point*. Sequences of configurations are also possible attractors; they can be repeated cyclically: then they are called *limit cycles*; or they can continue chaotically. More complex formations, where the evolution takes a little rest – possibly to mark something meaningful – and then goes on as mentioned above, can be *hyperbolic fixed points* attracting

from some directions and repelling towards others, or more complicated but more efficiently trapping *chaotic repellors* or semi-attractors (see e.g. Kantz and Grassberger 1985). Their role in the dynamics of neural networks has not yet been systematically studied.

The development reviewed in this book started by picking out one of the simplest computational tasks naturally demanding a solution by means of attractors: the associative (in other words: content–addressable) retrieval of stored memories.

A common computer stores information in files with a file name: that is the address of the memory. To have access to the information, you have to enter the address. Our brain proves to be a more intelligent computer on this task. It can do something better: a stored information can be found under many addresses; namely, those partial features of the information that are suitable to be associated to the whole.

In this scene the central memory to be retrieved is the attractor towards which our thinking converges. The set of ideas from which a stored memory can be recalled by association is called the *basin of attraction* of that memory. The more you learn about a thing, the broader its basin of attraction is in your brain. That of Pythagoras' theorem may include first the name of Pythagoras, then subsequently rectangular triangle, hypothenuse, length of a vector, mean square displacement in a random walk etc.

Many different stored items and their basins of attraction coexist in your brain. Associative retrieval actually consists of two steps: for a given initial state of mind (or: input) your brain has to choose by some criterion the most attractive of the stored memories, then follow the path of association to go to that most attractive memory. In this respect associative memory retrieval is also a prototype of all discrete optimization tasks, known to be computationally hard. Your brain performs remarkably well on this kind of tasks, and some people would be glad to teach the secret to their computers.

From the viewpoint of engineering applications ready to mimic whatever useful can be learned from even poorly understood principles of biology, associative memory is just a way to approach the extremely important task of *pattern*

recognition. To tell a first-rate product from a faulty one, or a submarine from a whale, is just a decision of whether a pattern of sensory inputs falls within the basin of attraction of a carefully devised sample.

Physics enters the story by the simple reason that the dynamics of physical systems used to have attractors. As described in more detail on the subsequent pages, some physical systems – simple magnets – are analogous to the nervous system in a more engineering respect (both can be regarded as consisting of strongly coupled two-state elements). If even the attractors are present, then we have a far-reaching physical analogy for thinking through attractors, in a beautiful unity between structure and function.

The tough point of the reasoning is that we can learn so many different things, and for some time it was not so easy to think of a physical system with so many attractors. Here came Hopfield's model which in a natural way arrives at just that subclass of magnetic systems: *spin-glasses*, which do exhibit a large number of attractors of their dynamic evolution.

Physics offers also a highly developed technical machinery to approach such models: this is *statistical mechanics*, with all its methods to handle complex systems consisting of a large number of simple elements in a noisy environment. As we shall see in the forthcoming pages, noise – just a nuisance for the average engineer – has a decisive, often indispensable role in the operation of neural networks.

Most of this book is about the blossoming wood grown out of the extremely vigorous grain of this analogy.

2 THE NEURON AND ITS MODEL

2.1 A quick look at the neuron

A part of a nerve tissue in the microscope looks quite irregular and structureless (figure 2.1). It was the Italian anatomist Camillo Golgi who invented how to stain such a tissue to give more contrast to its microscope image. By sheer good luck, staining showed something much more important: only well bounded parts of the tissue became stained, demonstrating that those parts are separated by membranes from the rest. Thus, the tissue consists of separate nerve cells, called *neurons*.

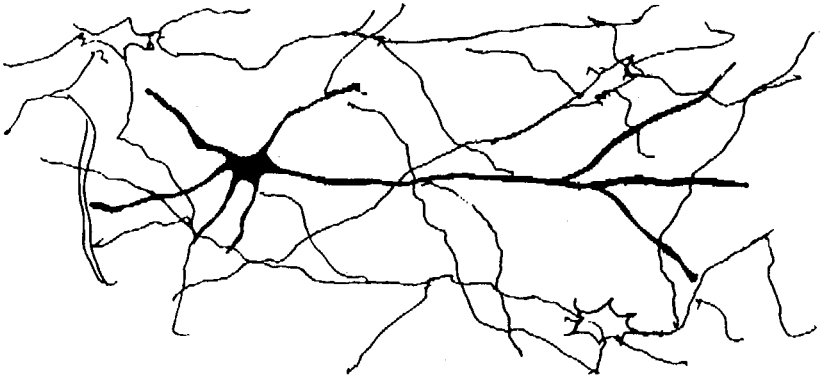


Figure 2.1. Part of a neural tissue with one neuron stained (schematic)

For the single neuron (figure 2.2) research has already reached the desired aim mentioned in the Foreword: the relationships between morphology and function are fairly well understood. The neuron is an elementary processor of information. Its body or *soma* takes care of the neuron's sustenance and does part of the processing; the short extensions on the soma, called *dendrites* are its input devices; the long extension or *axon* with all its branches is the output device. Finally the little thick foot on the end of each branch of the axon, touching another neuron (mostly on one of its dendrites), is the leading character of our story: a

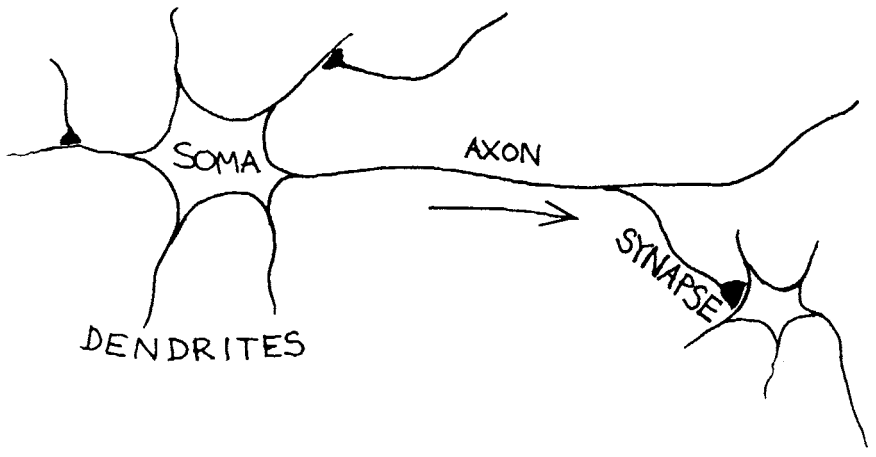


Figure 2.2 Parts of the neuron

synapse, a teachable, adaptable device of transferring information from one neuron to another. Adaptation of synapses is thought to be the basic step of memory storage in the nervous system.

How does all that work? Briefly: information is carried down the axon in the form of spike-like electric pulses, and transmitted through the synapse as chemical signals. The electric pulses are essentially invariable in shape and duration; information is encoded into their timing. It is an open question how accurate timing of the pulses is used for information coding in various parts of the nervous system; according to the simplest and best understood variant it is not the individual pulses, only their frequency that carries information. The frequencies of pulses incoming to a given neuron from various sources, modulated by the adaptable synapses, determine the frequency of outgoing pulses.

A little more detail about it: electric pulses arise on the background of a potential difference between the inside and the outside of the neuron, sustained by the action of molecular pumps driving sodium and potassium ions through the cell membrane. The inside of the cell is at the negative potential (typically about $-70\ mV$), which – usually measured by microelectrodes on the hillock

of the axon – changes under the influence of chemical signals. Whenever this inside potential rises above a sharply defined threshold value (about $-55\ mV$), the input signals loose control for a while: the potential flips up to a given positive value than back to negative again (figure 2.3), after which no such flip can be excited for some "refractory time". Starting with this potential pulse (or "action potential"), the neuron *fires*: the pulse runs down the axon, like a bullet, splitting into full pulses at each branching, to arrive at the synapses where they release some of the *transmitter* liquid contained in so-called synaptic vesicles (figure 2.4). The transmitter diffuses through the "synaptic slot" into the next neuron, in the membrane of which there are receptors converting the chemical signal back into changes of the electric potential.

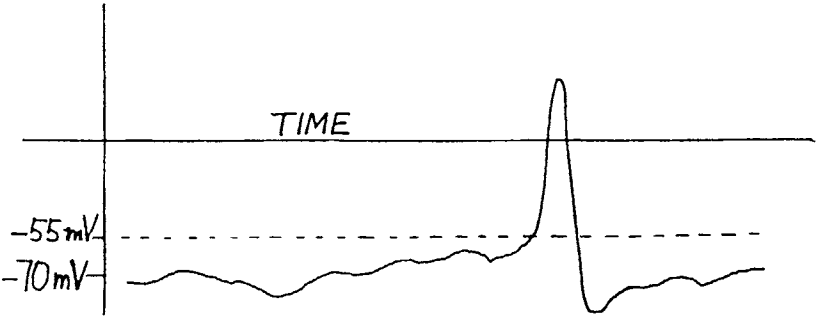


Figure 2.3 Time dependence of the electric potential difference between inside and outside the neuron, during a firing event

The potential change (of the order of $0.1\ mV$) induced by the chemical signal can be positive or negative, depending on the kind of the synapse. In the first case it contributes to enhance the firing frequency, therefore it is called an *excitatory* synapse. In the opposite case incoming signals reduce the firing frequency; then the synapse is called *inhibitory*.

Whether a given synapse is excitatory or inhibitory is fixed once for all: this cannot change on adaptation. Besides, all synapses carrying signals from a given

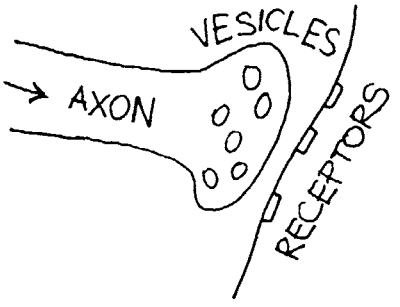


Figure 2.4 A closer look at the synapse

neuron are of the same type (all excitatory or all inhibitory), therefore neurons themselves can be classified as being either excitatory or inhibitory. In most of the existing models to be treated below these properties are not taken into consideration. Each success marked by such a model can then be regarded as a slight evidence that these properties concern more the "technology of fabrication" than the function. This may be in part misleading: at least in some cases excitatory and inhibitory neurons seem to carry profoundly different functions (see section 5.3).

The *size* of the (positive or negative) potential change due to an incoming pulse depends on the actual state of the synapse, which justifies to call it the *synaptic strength*. This is the property through which the neural network is open to teaching, adaptation and self-organization.

The changes of electrical potential caused by incoming pulses are retained and summed (integrated) during some time. If individual changes are small (synapses are not very strong) then this summation naturally acts in the direction that firing should depend mostly on the average frequencies of long trains of incoming pulses, not on their exact timing.

There is evidence of an opposite effect (Abeles 1982): pulses arriving *synchronously* through different synapses may excite a response much stronger than just the sum of individual responses. This ability of the neuron to detect coincidences is possibly the feature underlying the formation of the so-called *synfire chains:* well-defined sequences of individual pulses fired by a few coupled neurons. The way information is encoded into such patterns is as obscure for the moment as the writing of some extinct culture; however, the code seems to be used by our brain, at least in higher mental processes. In what follows we leave this exciting field of research and restrict our attention to the more pedestrian tool of the frequency code, doing apparently much of the rough work in the nervous system. However, the issue will be briefly taken up in sections 5.3 and 5.5.

The long-reaching axons with their synapses connect the neurons into a huge network. On this network *patterns* of firing and quiet neurons appear and change from time to time. Their changes, controlled by the teachable synaptic strengths, provide the processing of information that – at least so believe those of us with an inclination to reductionism – underlies most if not all mental processes.

2.2 Modelling the neuron

Our mathematical modelling is based on the assumption of an exclusive use of the frequency code: all information is carried by the firing frequencies of the neurons forming the network. Of course frequency itself makes sense only if neurons fire more or less evenly for some time. That makes the processing of information somewhat slow; in turn, it is much more robust against noise-induced shifts than individual pulses.

In this mode of operation it is the average frequencies of pulses incoming through excitatory and inhibitory synapses that determine how often the receiving neuron gets into the state of firing a pulse, i.e. what its frequency of firing is.

The output frequency y_i of the i-th neuron depends on the input frequencies x_j of pulses arriving from each neuron connected to the i-th one. This dependence comes about in two steps:

(i) Small changes of the time-averaged potential level z_i due to inputs arriving through different synapses add up in a smooth manner that one can apparently approximate by a linear function:

$$z_i = z_0 + \sum_j J_{ij} x_j \tag{2.1}$$

where z_0 is the level of rest. The weight J_{ij} is called the *synaptic strength* of the synapse on the i-th neuron, bringing input from the j-th one. These weights are the fundamental teachable, adaptable parameters of the neural network. J_{ij} is positive for an excitatory synapse and negative for an inhibitory one.

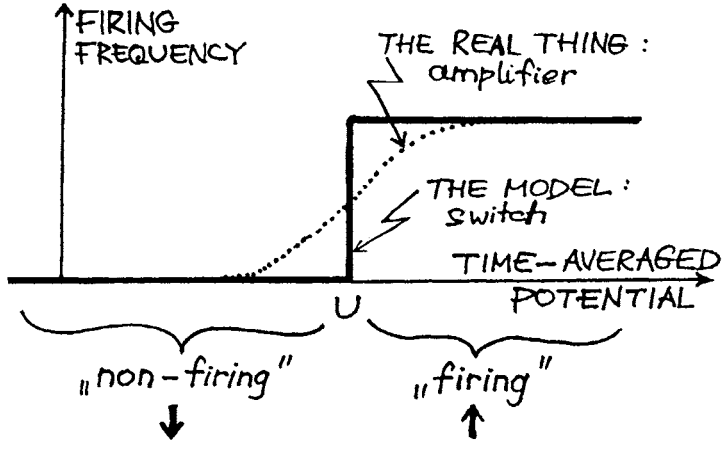


Figure 2.5 Firing characteristic of a neuron: frequency code *vs.* binary code

(ii) The average potential level determines the output firing rate in a very nonlinear fashion: a low potential level excites no firing at all, while even a high level cannot excite pulses in closer succession than the refractory time. That results in a functional dependence

$$y_i = f(\sum_j J_{ij} x_j - U), \tag{2.2}$$

where the non-linear transfer function (or "filter") $f(X)$ is of the "sigmoid" shape (figure 2.5). It has a characteristic step-like rising about the origin. U is a

threshold value for the average input level that is able to excite firing (related to, but not to be confused with the threshold potential for individual firing events). Sometimes it is convenient to view $f(X)$ as the characteristics of a very nonlinear amplifier.

We see that the single neuron is an efficient elementary processor, doing two operations in succession: a weighted summation of many inputs, followed by a non-linear filtering, based on comparison with a threshold. From a more abstract point of view such a processor is called a *threshold automaton.*

This one-neuron processor, as it stands, is already able to carry out a quite complex classification task: it can divide combinations of input arriving through different synapses into two classes, by giving high output to some combinations and low output to others. In this context the one-neuron processor is called a *perceptron.* Its modeling was an extremely important step towards neural computation (see Section 8.1).

A limiting case of a sigmoid $f(X)$ is an infinitely steep hard threshold: the Heaviside step function

$$f_\infty(X) = \Theta(X) = \begin{cases} 1 & \text{for } X > 0; \\ 0 & \text{for } X \leq 0. \end{cases} \tag{2.3}$$

In this limiting case each output is either zero or unity, thus we arrive at a *binary code*, as first suggested (for the oversimplified reason of "a pulse is either there or it is not") in the seminal paper of McCullogh and Pitts (1943), which marked the beginning of mathematical modelling of neural networks and gave a major impetus to von Neumann's work that culminated in building the first computers.

2.3 Feedback dynamical network

To build up the pulse trains of frequencies given by Eq. (2.2) takes some time after the inputs started. Imagine a neural network in which the output of each neuron goes to synapses on other neurons of the same network. Then the output frequencies become immediately inputs, which determine the outputs at a slightly later time.

Of course the pattern of outputs is usually not the same as that of the inputs was: the firing pattern changes in time. This defines the dynamics of activity changes of the network, which can be approximated as discrete-time dynamics:

$$V_i(t+1) = f\Big(\sum_{j=1}^{N} J_{ij}V_j(t) - U\Big) \quad (i = 1 \dots N), \tag{2.4}$$

where input and output firing rates, not principally different in the present case, were denoted by the same letters V_i $(i = 1 \dots N)$ that can be regarded as the elements of a time-dependent N-dimensional vector $\{V(t)\}$. This should be used for modelling mental processes that take place explicitly in time, like retrieving memories or taking decisions. On the other hand for highly automatic fast changes like visual processing in essentially feed-forward networks (Chapter 8) time has no explicit role and the simpler input-output relation (2.2) can be used.

In an important contribution Little (1974) achieved the impossible of combining the hard binary code with the soft-threshold characteristics of the sigmoid function (2.3). This could be done by re-interpreting $f(X)$ as the *probability* of getting zero or unit output for a given input.

In the development that followed it proved extremely useful to think in terms of a physical analogy: neurons, which are firing or quiescent, are analogous to the upwards or downwards pointing magnetic moments localized to atoms and attached to their "spins". Synaptic connections in this analogy correspond to physical interactions between the spins.

In this spirit, the binary variable $V_i = 1$ or 0 is now replaced by a spin variable $S_i = 2V_i - 1 = \pm 1$ $(i = 1 \dots N$ for N neurons or spins), familiar from the Ising model of magnetic systems: $S_i = +1$ (spin-up) if the i-th neuron fires at saturation rate, $S_i = -1$ if it is quiescent. The dynamical rule (2.4) is then interpreted as the random flipping of spin S_i under the summarized action of interactions with other spins and external magnetic fields. Indeed, the input to $f(\dots)$ can be re-written in terms of spin variables as $\frac{1}{2}\sum_j J_{ij}S_j(t) + \Big(\frac{1}{2}\sum_j J_{ij} - U\Big)$. To make contact with magnetic models easier, let us rescale the coupling strengths by just writing $\frac{1}{2}J_{ij}$ as J_{ij} (this is equivalent to use from now on twice smaller physical units to

measure it). Then the above expression becomes $\sum_j J_{ij}S_j(t)+\left(\sum_j J_{ij}-U\right)$. It is here that people with a physics background recognize the first sum as an "effective field" acting on spin i, originating from its interactions with other spins. On the other hand, the terms in the last bracket do not depend on the S_j-s: that is the "external magnetic field" that we write as h_i^{ext}.

With all this, the probability of spin-up in the next moment is the following:

$$S_i(t+1) = +1 \quad \text{with probability} \quad f\big(h_i(t)\big) \tag{2.5}$$

where

$$h_i(t) = \sum_j J_{ij}S_j(t) + h_i^{ext} \tag{2.6}$$

is the local field of two origines ("effective field" from interactions with other spins (neurons) and "external field" from thresholds and conversion from V- to S-variables).

Now comes statistical mechanics to offer a concrete and transparent form for the function $f\big(h_i(t)\big)$. A given noise level can be characterized by a "noise temperature" T (that has nothing to do with the true physical temperature of your brain!), the inverse of which is usually denoted by β. In a magnetic field h_i spin i has a potential energy $\epsilon_i = -h_iS_i$; accordingly, in the noisy ("finite-temperature") environment its probability to take a given value S_i is proportional to a Boltzmann-factor $e^{-\beta\epsilon_i} = e^{+\beta h_iS_i}$ that has to be properly normalized for the two possible values of S_i. This gives $f\big(h_i(t)\big) = e^{\beta h_i(t)}/\left(e^{+\beta h_i(t)} + e^{-\beta h_i(t)}\right)$. Slightly rearranging and calculating the complementary probability for $S_i(t+1) = -1$ too, the dynamical law appears as it is usually quoted:

$$S_i(t+1) = \pm 1 \text{ with probability } \frac{1}{1+e^{\mp 2\beta h_i(t)}}. \tag{2.7}$$

As can be easily checked, this function gives a larger probability for $S_i(t+1)$ to be parallel to the "local field" $h_i(t)$, but the difference between the two probabilities vanishes for a high noise level ($\beta \to 0$).

A neural network whose collective dynamics is described by Equation (2.7) is called the Little model (Little 1974). The Hopfield model, our main subject in this book, is obtained from it by some modifications to be described in Chapter 4.

For zero temperature (no noise; $\beta \to \infty$) the discrete-time dynamics is again deterministic:

$$S_i(t+1) = \mathrm{sgn}\Big(\sum_{j=1}^{N} J_{ij} S_j(t) + h_i\Big), \tag{2.8}$$

on which most studies of neural network dynamics are based.

What is achieved here reflects the particular attachment of physicists to symmetries: firing and non-firing, so much different from the biological point of view, are now just mirror images of one another, which – so we are trained by experience – simplifies the theoretical treatment quite a lot. Local asymmetries between the two sides are now lumped into the external fields h_i^{ext} that can be considered on a later stage of the analysis.

On the other hand, it should be born in mind that the use of equation (2.5) with its "V-variables" implies the presence of "external fields" strongly fluctuating from one neuron to the other, which may have pronounced effects (Bruce *et al.* 1987).

2.4 Connectivism

As mentioned at the end of Section 2.1, an elementary unit of the information handled by a neural network or a more or less autonomous part of it (say, an anatomically distinguishable part of a brain) is a spatial *pattern* of firing and quiescent neurons at a given time. The belief that the simultaneous firing states of the neurons carry some meaning together, not separately, rests on the observation that the connectivity of the network is extremely high: a single neuron can give pulses to and receive from some 10^3 to 10^4 others, which makes the dynamics of the system essentially collectively determined and delocalized. This is to be compared with a usual electronic circuit in which a transistor is connected to 3 or

4 others: there small parts of the circuit have considerable autonomy and evolve independently while other parts may do something else or take a rest.

The overall view that a strongly interconnected neural network and its firing patterns must be considered always as a whole became an important principle of orientation in the study of the nervous system; it is referred to under the name *connectivism*.

One further point should be mentioned here. Neurons in a brain are of different kinds, and some physiologists spend their lives refining and classifying the differences between them. In comparison, our models with just one species of neurons seem oversimplified indeed. An excuse can be that the number of different neurons is enormous with respect to the number of different kinds of them. In this way however important qualitative effects can be lost (in the same way as there is no plasma without positive and negative ions). At least the idea of distinguishing between active neurons of one kind and an infrastructural service background formed by "slave neurons" of some other kind returns from time to time (Peretto 1984, Kohonen 1988, Treves and Amit 1989).

3 HEBBIAN LEARNING

Now we want to show how a collective dynamical network as described in the previous chapter can store and retrieve information. Learning and memory is indeed a function of outstanding importance of our nervous system: not only our ability of adaptation to unforeseen environmental variations depends on it, but also as permanent tasks as visual processing are too complicated to be genetically fully encoded and must be acquired by learning in early infancy.

To proceed, it is time to make contact with Chapter 1 where we formulated some general expectations about the connection between memories stored in a network and attractors of its dynamics.

Although the firing pattern of an interconnected network of neurons usually changes in time, as modelled in the simplest noiseless case by the dynamical equation (2.8), some firing patterns may indeed turn out to be stable: one of them, say $S_i(t) = \xi_i$ $(i = 1 \dots N)$ once established, equation (2.8) gives it back for the next time step: $S_i(t+1) = \xi_i$ $(i = 1 \dots N)$, and so it remains unchanged indefinitely. In mathematical terms, the pattern ξ_i $(i = 1 \dots N)$, which can be also written as an N-dimensional vector $\{\xi\}$ of components $+1$ and -1, is a *fixed point* of the system of evolution equations (2.8).

Stability means actually more than that: the network may even turn back to the pattern after being forced out of it by some external disturbance. In other words, the pattern may turn out to be an *attractor* of the dynamics described by (2.8). The attractor has usually but a finite basin of attraction: from a very different initial pattern the network may approach another attractor, therefore instead of "stable" it is more accurate to say that the distinguished firing patterns are *metastable* states of the dynamical network.

Now attractors are just the thing needed for associative memory, as discussed in Chapter 1. Our task is indeed to encode a given piece of information into a firing pattern, then turn it into an attractor of the dynamics, preferentially with a considerable basin of attraction*.

* To specify patterns that should *not* belong to the basin of attraction is a

How can our brain do that? It is here that the numerous connections formed between the neurons acquire an active role. Although modifications of the shape of the transfer function – including the shift of thresholds – also have some chances, they do not seem to be a suffficiently flexible and still robust tool to entrust memory to them. It looks by far the most promising to change the numerous connection strengths J_{ij}; in their biological implementation: the synaptic strengths. This, in a non-mathematical way, was first discovered by Hebb (1948), who qualitatively specified the direction of the desired changes.

Hebb's idea was this: Let external inputs keep the memory network for some time in a pattern of firing that encodes some information to be stored. Storage is achieved if that pattern remains stable after removing the external input. If in the pattern both neurons i and j fire close to maximum rate and they are connected through an excitatory synapse, then this pattern has the more chance to become stable the stronger the synapse. Indeed, then other signals entering through other synapses have less chance to spoil this pattern. Therefore, the site of memory in the nervous system must be some biological mechanism by which excitatory synapses, when exposed to simultaneous firing on both their transmitter and receptor sides, get somewhat stronger.

To the reality of this mechanism convincing evidence has been found later, at least for simple animals (Kandel 1979). However, the idea may work in a broader setting. It would also act in the direction of stabilizing a pattern to inforce an inhibitory synapse J_{ij} if it brings strong firing from neuron j and neuron i is indeed quiescent, as required by this particular synapse. Of course the biological mechanism for that, if real, would be quite different, at least for the chemistry used.

With a bit of imagination you can think as well of a mechanism doing the reverse of the above two: to weaken a synapse whose message is frustrated (for j firing: i is quiet although J_{ij} is excitatory, or i fires although J_{ij} is inhibitory) would also do good service to memory.

---

problem of practical importance, so far not considered in the context of associative memory, although familiar in hetero-associative tasks, see Chapter 8.

However, the appetite of physicists for symmetry is inappeasable. Imagine (what is biologically hard indeed) that a synapse through which *no* firing arrives, would change according to whether the neuron just fires or not, and whether this satisfies or frustrates the idle synapse qualified as excitatory or inhibitory. Finally, what by now should not surprise you any more, assume that all these synaptic changes, irrespective of both their reality and the amount of satisfaction or frustration they would respond to, take place at the same intensity.

What is obtained is a remarkably simple formula for the modification of synaptic strengths when memorizing a pattern $\{\xi\}$:

$$J_{ij} \to J_{ij} + \lambda \xi_i \xi_j, \tag{3.1}$$

which is nowadays called *Hebb's rule*. The coefficient λ can be called the amplitude of learning; although biologically it *would* have some real value, physical modeling keeps it free for a convenient choice (see below).

Notwithstanding the enormous oversimplification in this formula, or maybe just because of that, Hebb's rule proved an extremely useful starting point for physical modelling of neural networks. The Reader interested in the complications arising from assigning different amplitudes to the biologically different processes it lumps together, should consult the paper of Peretto (1988).

If learning is started from zero connection strengths or *tabula rasa* and just one pattern is taught by Equation (3.1), the result $J_{ij} = \lambda \xi_i \xi_j$ is perfect, as can be immediately checked for the case of zero-temperature dynamics described by equation (2.8) restricted to a vanishing "external magnetic field" $h_i^{ext} = 0$:

$$S_i(t+1) = \mathrm{sgn}\Big(\sum_{j=1}^{N} J_{ij} S_j(t)\Big). \tag{3.2}$$

Indeed, for $S_i(t) = \xi_i$ the argument of the sign function is just $N\lambda\xi_i$, which gives back the same pattern for $t+1$.

The practical question is however more demanding: one has to store several patterns (say, p of them) on the same network. It is a miracle of connectivism that this is possible: apparently synaptic changes can be arranged in a way to make a

number of given patterns $\{\xi^\mu\}$ $(\mu = 1 \dots p)$ attractors of the network dynamics, each with its basin of attraction.

It is intuitively clear that the task cannot be solved for an arbitrary large number of patterns: there must be a maximum number of patterns stored without errors or with an acceptable fraction of errors (to be defined more precisely). This maximum number of stored patterns is in some sense a measure of the "storage capacity" of the network, and its determination is one of the main objectives of neural network modelling.

For this more complicated case Hebb's rule (3.1) still acts in the right direction, but the full task – if solvable at all – is expected to be solved by some iterative algorithm, taking one pattern at a time and eliminating errors by as little modification as possible to do little harm to the storage to the others, than proceed to another pattern and so on, until convergence. Indeed a number of such iterative algorithms exist and we will deal with them later on (see Sections 6.2 and 8.2).

There is however one exceptional case where the plain Hebb rule does the full job: the Hopfield model in its phase of "good memory". Now we turn to its study.

4 THE HOPFIELD MODEL

4.1 Definition of the model and basic facts about it

Models of coupled spins are familiar in the theory of magnetic phenomena. For modelling two-state neurons: firing or quiescent, we can use those spin models in which the spins cannot rotate freely, only point upwards or downwards: they are known as Ising models. Although largely oversimplified for real magnetics, they are often amenable for in-depth analysis by tools of statistical mechanics*, allowing one to understand collective phenomena which are often rather insensitive to the details of the elementary building blocks of a large system. Emphasizing this aspect of the problem is recently often advertised under the heading *Complex Systems*.

The recent uprise of interest of theoretical physicists in neural networks was triggered by Hopfield's observation (1982) that the interacting spin system analogy can give considerable insight to the mechanism of associative memory.

The underlying model is defined like this:

(i) Use the dynamic rule (2.8) for the noiseless or (2.7) for the noisy case, with the modification that instead of carrying out the required modification ("updating") of all spins from t to $t+1$ simultaneously (parallel dynamics), do it for one spin only at a time (serial dynamics). If that spin is chosen randomly at each time step, this is called Glauber dynamics.

(ii) Generate p patterns $\{\xi^\mu\}$ ($\mu = 1 \dots p$ is an upper index, not an exponent; remember that each pattern is an N-component vector with components $\xi_1^\mu \dots \xi_N^\mu$), each component being an independent random number of value $+1$ or -1. Fix connection strengths through Hebb-rule learning (3.1) started from *tabula rasa* (zero strengths), which results in

$$J_{ij} = \lambda \sum_{\mu=1}^{p} \xi_i^\mu \xi_j^\mu \quad (\text{for } i \neq j). \tag{4.1}$$

* The analysis is done most often by classical statistical mechanics, excluding the possibility of forming quantum superpositions of up and down spin states.

Self-coupling terms J_{ii}, if positive, tend to stabilize always the actual firing pattern and slow down the dynamical changes. However they have no influence on the attractors which are our primary concern here. For the moment we set $J_{ii} = 0$ which is advantageous for the statistical-mechanical treatment of equilibrium properties (see below).

The synaptic strengths obtained are apparently *symmetric*:

$$J_{ij} = J_{ji}. \tag{4.2}$$

If this symmetry property holds then it can be checked immediately that one can define an energy-like function of the firing pattern:

$$E\{S\} = -\frac{1}{2} \sum_{i,j(i \neq j)} J_{ij} S_i S_j - \sum_i h_i^{ext} S_i, \tag{4.3}$$

which can only decrease if a single spin is flipped in accordance with (2.8). Zero-temperature serial dynamics is a sequence of such single-spin flips: $E\{S\}$ will decrease then monotonously (serial dynamics has been chosen just for the sake of this property).

Such a monotonously decreasing function of the configuration is called a Lyapounov function of the dynamical system. If it exists, then its minima – local or global – are attractors towards which the system flows, until getting stuck in one of them.

This was associated by Hopfield to associative memory. Imagine the energy-like function $E\{S\}$ as a surface over the space of spin configurations (figure 4.1). This surface has minima at some configurations. Suppose that these minima correspond to the spin codes of stored memories or *patterns* ξ_i^μ ($i = 1 \ldots N$ neurons; $\mu = 1 \ldots p$ patterns). Then starting anywhere inside the potential well around a minimum, the spin configuration flows to the attractor: the stored pattern itself. Thus, in this visualization the basin of attraction appears as a potential well, and relaxation towards its bottom is just retrieving the stored pattern from its partial or distorted version.

On the other hand, for the physicist (4.3) is just the energy of a system of pairwise interacting Ising-spins $S_i = \pm 1$ in an external magnetic field. Such

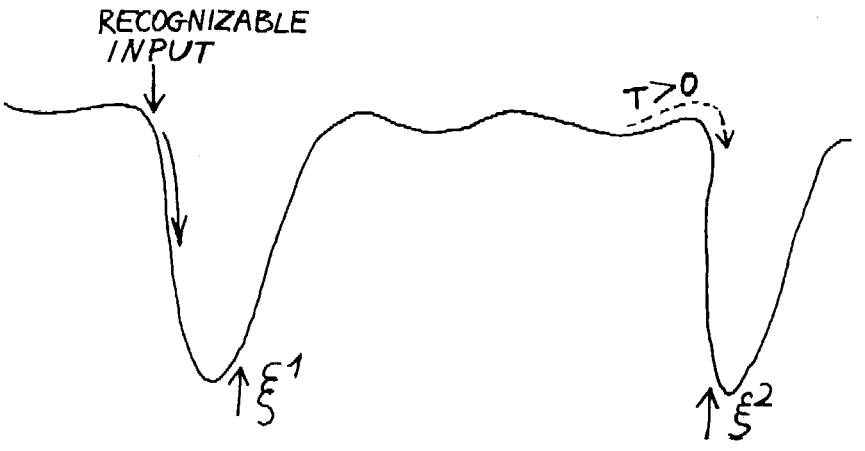


Figure 4.1 The quasi-energy surface of the Hopfield model: deep wells are attractors corresponding to stored patterns; shallow plateaus are spurious memories

systems used to undergo some kind of ordering as temperature gets low enough.

Speaking of coupled spin systems, a ferromagnet – at least its Ising-model carricature: a system of energy (4.3) with all coupling strengths J_{ij} positive – can have only two equilibrium configurations: all spins up or all down. Setting all coupling strengths negative does not help either: then (at least for spins on a cubic lattice) at low temperature we see a so-called antiferromagnet: another regular arrangement, with neighbouring spins alternating between up and down. Again, ups and downs can be interchanged: just two possibilities. How, then, can we learn and retrieve a large number of different things?

Here the analogy bursts into blossom. Since in the neural network there are both excitatory synapses with $J_{ij} > 0$ and inhibitory ones with $J_{ij} < 0$, this Ising model has a mixture of ferromagnetic and antiferromagnetic couplings. There are actually such real magnetic systems, called *spin-glasses* because spin orientations in them are frozen in an apparently random way, like atomic positions in window glass. Spin-glasses are an inexhaustible source of pleasure and headache for

physicists (for a recent review see Mézard *et al.* (1988), including an advanced theoretical treatment of the Hopfield model). A spin-glass has many equilibrium configurations because of the so-called *frustration* effect (Toulouse 1977): the irregularly placed negative couplings convey contradictory orientating forces to some of the spins, which therefore do not find one or two well-defined equilibria; instead, they find a number of poorly defined, frustrated equilibria. These, or at least part of them, correspond to the many different patterns you can store on a neural network.

How to force the system to have minima of $E\{S\}$ just at the configurations $\{S\} = \{\xi^\mu\}$, *i.e.* the pattern configurations to be stored? This is where the first and simplest guess: Hebb's rule (3.1) applied on tabula rasa, resulting in the coupling strengths (4.1), is tried and by miracle, it works under some conditions.

Like in Chapter 3, the simplest case is that of just one pattern $\{\xi\}$. Let us restrict ourselves to $h_i^{ext} = 0$. Then $E = -\frac{1}{2}\lambda(\sum_{i=1}^N \xi_i S_i)^2$, apart from an uninteresting additive constant. This is monotonously decreasing if the absolute value of $\sum_{i=1}^N \xi_i S_i$ (whatever its sign) grows, and stops at either $S_i = \xi_i$ or its mirror image $S_i = -\xi_i$ (for all i). This doubling of the pattern reflects the symmetry of the zero-field model with respect to reversing simultaneously all spins (referring back to the original meaning of spin, physicists used to call this time inversion symmetry).

Let us turn to the more interesting case of several patterns! To obtain some insight, the simplest thing to do is computer simulations. You can do it on any toy computer and it is amusing indeed. You can run the dynamics either with noise (equation (2.7)) and vary the inverse temperature β or without noise (equation (2.8)). You should try the model as *associative memory*, starting to run the dynamics from one of the stored patterns or close to it (with just a few errors) and see whether the pattern is recognized, i.e. whether the final configuration remains close to the starting pattern. Alternatively, you may start with a random initial state and watch if the system ends up close to one of the patterns or somewhere else.

The most important features, emerging the sharper the more neurons are put into the model and the more runs you average over, are this. The basic parameter governing the success or failure of the retrieval process is the ratio of the number of stored patterns p to that of neurons N. At $T = 0$, for $p \ll N$ associative retrieval is perfect. About $p \approx 0.1N$ a few errors begin to appear but not more than about 2%. Finally, on passing the famous limit of $p = 0.14N$ something very spectacular happens: the associative memory abruptly breaks down and patterns are not recognized any more. This overloading catastrophe very sharply defines the *storage capacity* of the model.

The explanation, to be supported first by a rough signal-to-noise estimate (Section 4.2) then by a more quantitative theory (Section 4.3), is briefly this: the different patterns disturbe each other's retrieval by creating some kind of static noise ("synaptic noise"); breakdown happens when this noise destabilizes the memories.

The case of non-associative, random retrieval is also very interesting. For noiseless, zero-temperature retrieval when starting from a random initial configuration, one practically *never* ends up in a stored pattern or its reversed. The reason is this: the quasi-energy $E\{S\}$ has a very large number of shallow local minima different from the stored patterns (see figure 4.1): these are the so-called *spurious memories*. Although the true patterns are there and their potential wells are deeper, it has a very large probability that one of the spurious memories catches the system.

The energy surface picture offers the remedy, which is a central claim for physicists' presence in the field of neural network modelling. You have to add *noise*: take some finite temperature (try how big is best); then the system gets a chance to move occasionally upwards in quasi-energy and climb out of the dolina of the spurious memory, to find its way to one of the true memories. This is beneficial noise: a stranger to usual engineering!

Temperature should not be too high however: spurious memories, although higher in energy, are much more numerous than stored patterns: for their number $n_{sp} \gg p$ holds, and even a small thermal occupation probability $\propto e^{-\beta E_{sp}}$ for

each single one of them may give a large probability $\propto n_{sp}e^{-\beta E_{sp}} = e^{-\beta F_{sp}}$ for the system to remain in anyone of them, where $F_{sp} = E_{sp} - TS_{sp}$ can be regarded as the (quasi) free energy of the set of spurious memories, $S_{sp} = \ln n_{sp}$ being their entropy and $T = \beta^{-1}$ the temperature. Thus, it is always the subset of lower free energy at a given noise temperature which attracts the dynamical flow more efficiently: those which are more numerous for high T, those which are lower in energy for low T; except if too low noise temperature kinetically inhibits thermalization.

This reasoning points to a great potentiality of the Hopfield model: although the energy-like function has nothing to do with the true physical energy of a human brain or an electronically implemented neural net, the model can be treated by the familiar tools of *equilibrium* thermodynamics and statistical mechanics. In particular, e.g. calculating the quasi free energy and searching its minima, one obtains reliable information about the ranges of noise level – both "thermal", associated to retrieval dynamics, and "synaptic", due to the presence of many stored patterns – where good memory operation can be expected. This is the principal reason why Hopfield's model, with all its oversimplifications and practical drawbacks, became the "origin of the coordinate system" for physicists working in neural network modelling. This topic will be treated below in Section 4.3.

4.2 Noise analysis

To have a somewhat more quantitative feeling about how the breakdown by synaptic noise happens, let us introduce the *overlap* of the actual configuration with pattern μ:

$$m_\mu = \frac{1}{N}\sum_{i=1}^{N} S_i \xi_i^\mu. \tag{4.4}$$

Let us notice that $d_\mu = N(1 - m_\mu)/2$ is the so-called *Hamming-distance* between the configuration and pattern μ: the number of bits in which they differ from each other.

With the overlaps m_μ the quasi-energy (4.3) for the coupling strengths (4.1) (self-coupling excluded) and zero external fields can be written in the form

$$E = -\frac{\lambda}{2} N^2 \sum_{\mu=1}^{p} (m_\mu)^2 + \frac{\lambda}{2} Np. \tag{4.5}$$

Let us assume now that pattern ν or its (time-)reversed is retrieved perfectly or almost so: then $(m_\nu)^2 = \mathcal{O}(1)$. If the other terms for $\mu \neq \nu$ were not there, everything would happen like for a single pattern.

Now it is the peculiarity of Hopfield's model that the other terms, representing what we called above the synaptic noise, are indeed *individually* small. The sense of defining the model with random independent patterns stored in the coupling strengths was just to assure this property. Then if the configuration is parallel or antiparallel with one pattern, the overlaps with the others are sums of N random numbers $+1$ or -1, summing up to $\mathcal{O}(N^{-1/2})$ which is small for large N.

However small the noise terms, if they are numerous, they can add up to be comparable with the $\mathcal{O}(1)$ "signal" term: that is when the associative recall of pattern ν becomes impossible. Of course none of the patterns enjoys a distinguished status; whenever the Hopfield network is overloaded, each pattern is spoiled by all the others: that is the famous breakdown.

Actually what is needed for associative memory operation is not just random patterns, only that the stored patterns should be essentially *orthogonal*, which means that their scalar-product-like *overlap* defined as

$$q_{\mu\nu} = \frac{1}{N} \sum_{i=1}^{N} \xi_i^\mu \xi_i^\nu \tag{4.6}$$

should vanish for $N \to \infty$. The case of independent random binary elements ± 1 satisfies this requirement, since their overlap is $\mathcal{O}(N^{-1/2})$, although it is just this slight non-orthogonality of random patterns that causes the synaptic noise. On the other hand, patterns with a *bias*: $\sum_i \xi_i^\mu = \mathcal{O}(N)$ are not orthogonal according to the above definition and cannot be reliably distinguished from each other by the plain Hopfield model. This is a serious practical limitation of the model and

much effort has been devoted to get rid of it. We will return to the problem in Chapter 5.

The simple signal-to-noise analysis can be driven further on, which is the reason it still continues to give useful first orientation about novel variants of neural network models with associative memory. The aim is to calculate how much the retrieval of a given pattern is disturbed by the presence of other stored patterns.

We restrict ourselves to noiseless dynamics: $T = 0$. Then for Hopfield's model errorless retrieval of a pattern $\{\xi^\nu\}$ means that it should be a fixed point of the dynamics (2.8), *i.e.* $\xi_i^\nu \sum_{j \neq i} J_{ij} \xi_j^\nu$ should be non-negative for all i.

A large positive value of this sum is intuitively connected to a large basin of attraction for the ν-th pattern (cf. Chapter 6), *i.e.* a large stability of the pattern configuration against errors. To enhance this stability, one has two ways; one trivial, the other not. The trivial is to multiply each J_{ij} by the same constant; the non-trivial is to improve learning by a more careful adjusting of the different coupling strengths. To exclude the trivial effect – which is trivial only to say, biologically or electronically quite implausible – from the forthcoming considerations, let us introduce the normalized quantity

$$\Delta_i^\nu \equiv \frac{\xi_i^\nu \sum_{j \neq i} J_{ij} \xi_j^\nu}{\sqrt{\sum_{j \neq i} J_{ij}^2}}. \tag{4.7}$$

This is the combination usually called the *stability* of pattern $\{\xi^\nu\}$ at the i-th neuron. Then the condition of errorless retrieval is $\Delta_i^\nu > 0$ everywhere.

For random patterns it is easy to give an accurate upper bound for p up to which this condition is obeyed *with finite probability* for $N \to \infty$ (Weissbuch and Fogelman-Soulié 1985).

The derivation starts by calculating the probability ε of error in a single bit, *i.e.* that of violating the positivity requirement for some given neuron i. From the definition (4.7) with the connection strengths (4.1), approximating the normalizing denominator by its mean $\lambda\sqrt{p(N-1)}$ (which is exact for $N \to \infty$ since its dispersion is only $\lambda\sqrt{p-1}/2$), we have

$$\Delta_i^\nu = \sqrt{\frac{N-1}{p}} + \frac{1}{\sqrt{(N-1)p}}(+1+1-1+1-1\ldots.), \tag{4.8}$$

where the sum of $(N-1)(p-1)$ (corresponding to all $j \neq i$, $\mu \neq \nu$) independent random variables ± 1 approaches a Gaussian variable of zero mean and $\sqrt{(N-1)(p-1)}$ dispersion. Then with the factor in front of the random sum, for $N \gg 1$ and $p \gg 1$ we obtain

$$\Delta_i^\nu \approx \sqrt{\frac{N}{p}} + z \tag{4.9}$$

z being a Gaussian random variable of zero mean and unit dispersion. The desired probability that Δ_i^ν is negative is obtained by integrating its probability distribution: $\varepsilon = \int_{-\infty}^{0} d\Delta_i^\nu \mathcal{P}(\Delta_i^\nu)$, *i.e.*

$$\varepsilon = \frac{1}{2}\mathrm{erfc}\sqrt{\frac{N}{2p}} = \frac{1}{2}\mathrm{erfc}\sqrt{\frac{1}{2\alpha}} \tag{4.10}$$

where $\mathrm{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$ is the complementary error function (Abramowitz and Stegun 1972) and

$$\alpha \equiv \frac{p}{N}. \tag{4.11}$$

Using the known asymptotics $\mathrm{erfc}(x) \sim e^{-x^2}/\sqrt{\pi}x$ for $x \to \infty$, we obtain

$$\varepsilon \approx \sqrt{\frac{\alpha}{2\pi}} e^{-1/2\alpha} = \sqrt{\frac{p}{2\pi N}} e^{-N/2p} \tag{4.12}$$

for $\alpha \to 0$, the limit in which errorless retrieval is expected. Notice that $\alpha = 0$ is a point of essential singularity where $\varepsilon(\alpha)$ vanishes with all its derivatives.

At this point we need an argument to regard Δ_i^ν and Δ_l^ν independent random variables for $i \neq l$. Indeed, they are sums of independent terms (see equation (4.7)), apart from the term $J_{il}\xi_i^\nu \xi_l^\nu$ which is common for the two. However, for a model with *macroscopic connectivity*, in which each spin (neuron) is connected to infinitely many others for $N \to \infty$, the contribution of this single common term is negligible with respect to the many independent ones. This argument is

strongly reminiscent of the use of mean-field theories for systems with long-range interactions.

Making use of this independence, the probability of no error is given by

$$P_0 = (1-\varepsilon)^N, \tag{4.13}$$

which should remain finite for $N \to \infty$, according to our starting point. Taking the logarithm, *then* expanding in $\varepsilon \ll 1$ and substituting ε from (4.12), one obtains

$$2\ln N - \frac{N}{p} - \ln\frac{N}{p} = \ln\bigl(2\pi(\ln P_0)^2\bigr). \tag{4.14}$$

According to our starting assumption, the right-hand side should remain finite for $N \to \infty$. Then the last term on the left-hand side is negligible and we obtain the upper bound below which the requirement can be satisfied:

$$p \le \frac{N}{2\ln N}. \tag{4.15}$$

It is easy to obtain lnln corrections to this asymptotic bound.

The above result means that although p can grow with N, however α should vanish as $(2\ln N)^{-1}$ if errorless retrieval is required.

The extension of the above probability reasoning to the case of finite $\alpha = p/N$ (Hopfield 1982, followed by many workers) is problematic but useful as a first check if one does not ask too ambitious questions. What one can calculate in this way is the probability distribution of the fraction of erroneous (unstable) bits *in the pattern configuration* $\{\xi^\nu\}$. By the mean-field-like argument used before Eq. (4.13), this will be a Bernoulli binomial distribution. Using its known mean and dispersion it is easy to see that for large N the distribution is sharply peaked and the error fraction is just ε, with accuracy $N^{-1/2}$.

However, what we calculated is only the fraction of bits that begin to flip away if we *start* from the pattern configuration, asking whether the memory recognizes the pattern. The retrieval error, instead, is given by the distance between the pattern and the *final* state of the relaxation. According to the experience

mentioned above, for α just below the critical value $\alpha_c = 0.14$, where the pattern-configuration error can be calculated from (4.10) to be 0.4%, the final-state error grows by flipping-away to 1.5% only. However, for slightly more error in the initial configuration, the relaxation runs away to no recognition of the pattern at all.

It is important to bear in mind that the above simple analysis gives no clue as to where this breakdown takes place, and a detailed analysis of how the dynamics of retrieval proceeds for several time steps is a complicated subject (see Chapter 7). Signal-to-noise analysis is still a useful tool to estimate the place of the breakdown in variants of the model, by calculating where the one-bit error in the pattern configuration would reach the critical value $\varepsilon_c = 0.004$. On a later stage of the study however such an estimate has to be supported by a mean-field-type calculation (see below).

The view that it is the overall noise level that determines the evolution in the basin of attraction of a single pattern, was taken up in a more quantitative dynamical calculation by Krauth *et al.* (1988). A limitation is that if learning amplitudes of the different patterns are unequal, a strong pattern may destabilize the weaker ones much before noise rises to a dangerous level (Pázmándi and Geszti 1989). This development will be described in more detail in Section 7.2 and Appendix B.

The attractor-centric view at neural networks tends to mask that the simple noise analysis is exact for the first step of dynamics, and in the first step there is no breakdown at $p = 0.14N$. Started in a might-be basin of attraction, even a strongly overloaded network moves one step towards the stored pattern, only later steps reveal that a true attractor is not there. This "dynamical retrieval" in an overloaded network (Gardner *et al.* 1987, Fontanari and Köberle 1988b) is one of the practical potentialities of Hopfield networks. Large positive self-couplings, keeping the network for a long time close to where it moved at first, improve this kind of performance.

4.3 Mean-field theory of the Hopfield net: derivation of the Amit-Gutfreund-Sompolinsky equations

As sketched near the end of Section 4.1, the relative probabilities that the system under "thermal" noise would stay in one or another of its attractors – true stored pattern or spurious memory – can be calculated from the statistical mechanics of thermal equilibrium: it is Boltzmann factors $e^{-\beta F}$ associated with a quasi free energy F which determine these probabilities.

In order that the apparatus of statistical mechanics should not give discomforting results when taking the thermodynamical limit $N \to \infty$, the quasi-energy $E\{S\}$ should have a property familiar with true physical systems: it must be an *extensive* quantity, *i.e.* for a large system it should be proportional to the number N of neurons (spins). This is satisfied without saying in a network where each neuron is connected to a fixed number of its neighbours only. The Hopfield model is defined however with each neuron being connected to all other neurons ("fully connected network"). Therefore in equation (4.5) E appears with a factor λN^2, *i.e.* it would grow proportionally to the number of connections, not of neurons.

The difficulty can be eliminated by fixing the "learning amplitude" to $\lambda = 1/N$. This is no real restriction, since in equilibrium statistical mechanics the energy scale factor λ appears always in the Boltzmann factors containing the combination $\beta\lambda$, therefore a change of λ is equivalent to rescaling the inverse noise temperature β. In particular, theory says that the interesting things (*e.g.* destabilization of retrieval by too much noise) will happen at a noise level $\beta^{-1} = \mathcal{O}(\lambda N)$, in the present rescaling however this will be described by size-independent critical temperatures of $\mathcal{O}(1)$ (see Section 4.4).

With the above choice, equation (4.5) gives for the energy per spin

$$\frac{E}{N} = -\frac{1}{2}\sum_{\mu=1}^{p}(m_\mu)^2 + \frac{\alpha}{2}, \tag{4.16}$$

where the familiar notation $\alpha = p/N$ has been used.

By this we have obtained a powerful insight into the action of quasi-thermal noise on the associative memory operation of *large* networks ($N \gg 1$). Indeed, the

configurations qualitatively different for the retrieval process (in a thermodynamic language: different "phases" of the network) are those for which the overlaps m_μ differ by numbers of order unity. Such configurations however, according to (4.16), differ strongly in energy: $\Delta E \propto N$. The same is true for free energy differences. Therefore at a noise level of order unity the Boltzmann factor ratios $e^{-\beta \Delta F}$ are very large or very small depending on the sign of ΔF, and transitions between different phases, switching to the one of lower free energy, happen with the familiar sharpness of a thermodynamical phase transition. Moreover, free energy barriers separating local minima of the free energy are also $\mathcal{O}(N)$, therefore a dynamically selected phase (*e.g.* a given stored pattern located by associative retrieval), corresponding to a local minimum of free energy, is retained with probability close to 1 for a very long time, in spite of the presence of noise.*

The above reasoning was put on a sound quantitative basis in the fundamental paper of Amit, Gutfreund and Sompolinsky (1987), hereafter referred to as AGS. They carried out a mean-field calculation using the replica trick familiar from the theory of spin-glasses (Kirkpatrick and Sherrington 1978). The method is now so widespread that one is unable to follow the literature without getting some familiarity with it. Therefore the calculation is reviewed in Appendix A of the present book. However, to get a quick feeling about the physical content of mean-field theory applied to the Hopfield model, we present here a simple approximate derivation, found independently by Peretto (1988a).

The calculation starts from the system of Weiss mean-field equations familiar from the elementary theory of ferromagnetism. To derive them, let us take the "thermal average" of the two possible values of $S_i(t+1)$ according to equation (2.7) describing noisy dynamics. Denoting this average by angular brackets and for simplicity omitting external fields, the result is

* The presence of $\mathcal{O}(N)$ barriers places the retrieval process in one class with computationally hard combinatorial optimization problems to which carefully tuned noise level schedules ("simulated annealing") are known to offer a powerful approach (Kirkpatrick *et al.* 1983).

$$\langle S_i(t+1)\rangle = \tanh\Big(\beta \sum_{j=1}^{N} J_{ij} S_j(t)\Big). \tag{4.17}$$

We want to describe retrieval states of the noisy network, *i.e.* stationary states *on the average*:

$$\langle S_i\rangle = \Big\langle \tanh\Big(\beta \sum_{j=1}^{N} J_{ij} S_j\Big)\Big\rangle \tag{4.18}$$

where on both sides averaging is taken over the stationary distribution of the individually fluctuating spin variables.

The mean-field approximation to this equation is to replace the average of the non-linear tanh function of spin variables on the right-hand side by the tanh function of the averaged spins:

$$\langle S_i\rangle = \tanh\Big(\beta \sum_{j=1}^{N} J_{ij} \langle S_j\rangle\Big). \tag{4.19}$$

This would be exact if the argument of the tanh function were sharply determined. The approximation is based on the expectation that although the individual spins are strongly fluctuating, the argument of the tanh is a sum of $\mathcal{O}(N)$ terms and this may turn out to have a much smaller dispersion. This is indeed exactly satisfied for a ferromagnet of infinite-range interaction. For a disordered system the situation is more complicated; Mézard *et al.* (1988) describe a systematic approach called the "cavity method" to handle deviations from the simple mean field. Here we take equation (4.19) as an approximation and go on with the analysis to see the consequences.

(4.19) is a closed system of equations for the thermally averaged spins $\langle S_i\rangle$. For the retrieval problem it is advantageous to switch to new variables: the overlaps of the spin configuration with the patterns. These overlaps have been introduced in equation (4.4); now we redefine them as thermal averages:

$$m_\nu \equiv N^{-1} \sum_j \xi_j^\nu \langle S_j\rangle, \tag{4.20}$$

which are analogous to the average magnetization of a ferromagnet. Then introducing the Hebb-rule connection strengths (4.1) with $\lambda = N^{-1}$, multiplying equation (4.19) by ξ_i^ν and summing over i, we obtain

$$m_\nu = N^{-1} \sum_i \tanh(\beta \sum_\mu \xi_i^\nu \xi_i^\mu m_\mu), \tag{4.21}$$

which is the starting point for our following analysis.

The physical idea of the calculation (of course triggered by the replica results) is to distinguish between large (retrieved) and small (noise) overlaps and approximate the latter by a noise of given statistical distribution whenever possible, however to carefully take into account the fact that the small overlaps themselves are also thermal equilibrium values, determined by the mean-field equations.

In what follows we assume that the noise to which the small components add up is Gaussian. This is an oversimplification since the small components contributing are connected by the mean-field equations and therefore not independent. Most probably (Mézard *et al.* 1988, Kondor 1988) this is the step responsible for obtaining below the results corresponding to the replica-symmetric approximation in the AGS paper (see the Appendix A).

In this spirit, let us regard the case of a single large overlap $m_1 = \mathcal{O}(1)$, i.e. the retrieval of a single pattern $\{\xi^1\}$. This *ordering* of spins along one of the patterns is strictly analogous to ferromagnetic ordering in one of the possible directions; m_1 is the *order parameter* characterizing the degree of perfection this ordering took place. The equation expected to determine it is

$$m_1 = N^{-1} \sum_i \tanh \beta(m_1 + \sum_{\mu \neq 1} \xi_i^\mu \xi_i^1 m_\mu). \tag{4.22}$$

The Gaussian noise approximation is applied now to the last sum in the bracket. For independent random patterns its dispersion is $\sum_{\mu \neq 1} m_\mu^2$. Taking into account the fact that for $p \gg 1$ the restriction $\mu \neq 1$ gives only a negligible $\mathcal{O}(p^{-1})$ change of the noise amplitude, and that terms with different i differ only in the random signs of $\xi_i^\mu \xi_i^1$, therefore averaging over the Gaussian noise takes

care of the site averaging $N^{-1}\sum_i$ as well (which is a much weaker statement than "self-averaging"), we obtain

$$m_1 = \int \frac{dz}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} \tanh\beta(m_1 + \sqrt{\alpha r}z), \tag{4.23}$$

where the random overlap parameter r is defined through

$$r = \frac{1}{\alpha}\sum_{\nu\neq 1} m_\nu^2. \tag{4.24}$$

This overall characteristics of synaptic noise is regarded as a second order parameter of the problem. It comprises two effects: random overlaps between the patterns themselves, and weak condensation of the spin configuration along the non-retrieved patterns.

The roughest estimate would neglect the second thing and assume that $m_\nu^2 \approx N^{-1}$; then for $p \gg 1$ patterns one has $r = 1$. Such a treatment would give a second-order phase transition with growing α (Kinzel 1984). Here, instead, we take into account that small overlaps are also subject to the requirements of thermal equilibrium, and determine r from Eq. (4.21) applied for a small component m_ν, $\nu \neq 1$:

$$m_\nu = \frac{1}{N}\sum_i \xi_i^\nu \xi_i^1 \tanh\beta(m_1 + \sum_{\mu\neq 1,\nu} \xi_i^\mu \xi_i^1 m_\mu + \xi_i^\nu \xi_i^1 m_\nu). \tag{4.25}$$

The crucial step of the derivation is to linearize in the last term in the bracket, which gives a coherent (non-noisy) contribution, analogous to the Onsager reaction field contained in the cavity method (Mézard *et al.* 1988):

$$\begin{aligned} m_\nu \approx & N^{-1}\sum_i \xi_i^\nu \xi_i^1 \tanh\beta(m_1 + \sum_{\mu\neq 1,\nu} \xi_i^\mu \xi_i^1 m_\mu) \\ & + \beta N^{-1}\sum_i (1 - \tanh^2\beta(m_1 + \sum_{\mu\neq 1,\nu} \xi_i^\mu \xi_i^1 m_\mu)) m_\nu. \end{aligned} \tag{4.26}$$

Expressing m_ν and applying again the reasoning before equation (4.23), we obtain

$$m_\nu = \frac{N^{-1}\sum_i \xi_i^\nu \xi_i^1 \tanh\beta(m_1 + \sum_{\mu\neq 1,\nu} \xi_i^\mu \xi_i^1 m_\mu)}{1-\beta(1-q)}, \tag{4.27}$$

where

$$q = \int \frac{dz}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} \tanh^2\beta(m_1 + \sqrt{\alpha r}z) \tag{4.28}$$

is another order parameter of the problem (the third after m_1 and r), familiar from the theory of spin glasses: the Edwards-Anderson order parameter. Its physical content is this: the average of $(\tanh)^2$ measures the extent the spin system is frozen into *any* configuration, whether parallel to a pattern or not.

In view of the random factors in front of the tanh in (4.28), the Gaussian averaging could not be done in the numerator in this step.

Now we proceed to calculate the "random overlap parameter" r defined in Equation (4.24). For this we need

$$\begin{aligned} m_\nu^2 =& N^{-2}(1-\beta(1-q))^{-2} \sum_{i,j} \xi_i^\nu \xi_i^1 \xi_j^\nu \xi_j^1 \times \\ & \tanh\beta(m_1 + \sum_{\mu\neq 1,\nu} \xi_i^\mu \xi_i^1 m_\mu) \tanh\beta(m_1 + \sum_{\mu\neq 1,\nu} \xi_j^\mu \xi_j^1 m_\mu). \end{aligned} \tag{4.29}$$

Since ξ_i^ν is excluded from the arguments of the tanh's, the outside four-ξ factor can be averaged independently, to give a δ_{ij}. Then we obtain

$$N m_\nu^2 = (1-\beta(1-q))^{-2} N^{-1} \sum_i \tanh^2\beta(m_1 + \sum_{\mu\neq 1,\nu} \xi_i^\mu \xi_i^1 m_\mu), \tag{4.30}$$

where the Gaussian noise approximation can be taken now, like in the derivation of Equation (4.27). Then the right-hand side of (4.30) becomes independent of the index ν, and the final result, using Equations (4.24) and (4.28), is

$$r = \frac{q}{\left(1-\beta(1-q)\right)^2}. \tag{4.31}$$

Equations (4.23), (4.28) and (4.31) are identical to the mean-field equations for the Hopfield model determining the three order parameters m_i, r and q, as first obtained in the replica symmetric approximation by AGS and described here in the Appendix.

4.4 Mean-field theory: solution of the equations

The information contained in equations (4.23), (4.28) and (4.31) is discussed in much detail in the original paper of AGS that the reader seriously interested in the subject cannot miss to work through. Here we discuss the simplest features only. For a single "macroscopic" ($\mathcal{O}(1)$) overlap we simplify the notation to $m_1 = m$.

For zero noise temperature ($\beta \to \infty$) the integrals can be simplified by noting that for $a \to \infty$ $\tanh(ax) \to \mathrm{sgn}x$ and $(\tanh)^2(ax) \to 1 - \frac{2}{a}\delta(x)$, where $\delta(x)$ denotes the Dirac delta function. Then from equation (4.23), using the oddness of the integrand, we obtain

$$m = \mathrm{erf}\frac{m}{\sqrt{2\alpha r}}, \tag{4.32}$$

while (4.28) simplifies to

$$q = 1 - \frac{1}{\beta}\sqrt{\frac{2}{\pi\alpha r}}\exp\left(-\frac{m^2}{2\alpha r}\right), \tag{4.33}$$

from which it can be immediately seen that whatever r, at zero temperature $\beta \to \infty$ we obtain for the Edwards-Anderson order parameter $q \to 1$ expressing the obvious fact that without noise a stationary state is at the same time a completely frozen state.

With equation (4.31) an apparent ambiguity still remains: the product $\beta(1-q)$ is undetermined for $\beta \to \infty$. Its limit can be expressed however through m and r from (4.33), which substituted into (4.31) with $q = 1$ gives

$$\sqrt{r} = 1 + \sqrt{\frac{2}{\pi\alpha}}\exp\left(-\frac{m^2}{2\alpha r}\right). \tag{4.34}$$

Dividing (4.32) by this last equation, we obtain a closed equation for the single variable $y = m/\sqrt{2\alpha r}$:

$$y = \frac{\mathrm{erf}(y)}{\frac{2}{\sqrt{\pi}} e^{-y^2} + \sqrt{2\alpha}}. \tag{4.35}$$

This equation has always a solution $y = 0$. However for $\alpha < \alpha_c = 0.138$ other pairs of solutions (reverse to each other) exist as well (figure 4.2), and one pair of them is locally stable: they attract the dynamics from initial configurations of a macroscopic overlap with the pattern to be retrieved. They correspond to local minima of the free energy (see Appendix A).

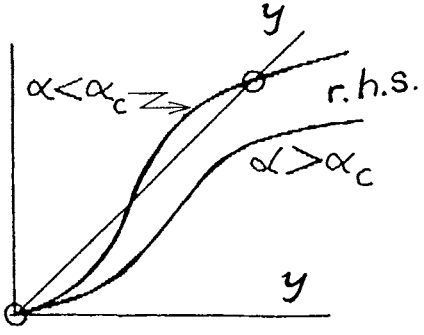


Figure 4.2 Solution of Equation (4.35) (schematic)

A stable solution with $m \neq 0$ corrresponds to a *retrieval state*, corresponding to associative memory action. The pattern retrieved is not exactly the stored one but close to it: the mean error fraction is $x = (1 - m)/2$. Errorless retrieval apparently allows a much smaller number of stored patterns (Section 4.2) or desires a learning algorithm more sophisticated than the simple Hebb rule (3.1) (Chapter 6). However, retrieval with an acceptable fraction of errors has its biological attraction.

From the solution of (4.35) we obtain $m = \mathrm{erf}(y)$. At α_c its value is $m_c = 0.967$ which corresponds to an error fraction $x_c = 0.016$. Above α_c there is no solution but $m = 0$, and the error fraction jumps up abruptly to the random value 0.5.

The solution with $m = 0$ and $q = 1$ corresponds to a *spin-glass* state: the spin configuration is frozen but orthogonal to all patterns, *i.e.* looks irregular with respect to each of them.

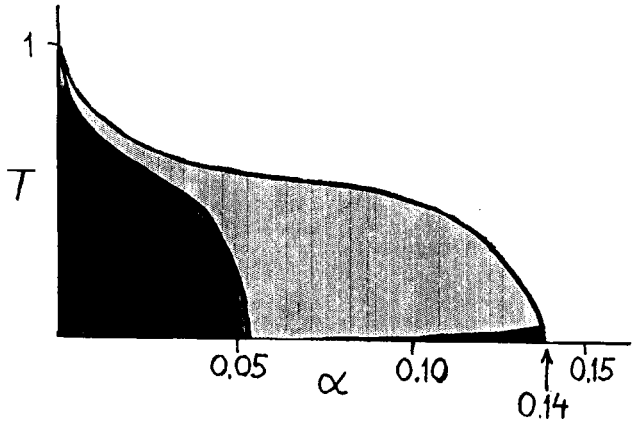


Figure 4.3 The Amit-Gutfreund-Sompolinsky (AGS) phase diagram (grey: retrieval states with less than 1.5% of errors are metastable with a large basin of attraction; dark grey: retrieval states are globally stable; black: retrieval with replica symmetry breaking)

Finite level of noise is a major issue in physical modelling of neural networks. The mean-field equations can be solved numerically for a finite temperature as well. The result is the famous AGS phase diagram (figure 4.3), indicating the range in pattern number and noise temperature where the stored patterns – modified by an average fraction $(1-m)/2$ of errors – are attractors, therefore associative memory is expected to work. Such phase diagrams are the useful thing for artificial neural networks too: in electronic implementations of neurons the gain of an amplifier plays a role analogous to the inverse noise temperature β (big gain *i.e.* sharp switching corresponds to low noise: Hopfield 1984). Therefore the phase diagram gives a hint to the electronic parameters required. Computer simulations on the Hopfield model give a detailed justification of the theoretically obtained phase diagram.

On the low-α side of the phase diagramm new attractors appear, neither of the retrieval, nor of the spin-glass type. These are the *spurious states* already mentioned in Section 2.1. They are mixtures of a few of the stored patterns. The easiest way to study them is to solve equation (4.23) for $\alpha \to 0$, e.g. a finite number of stored patterns while $N \to \infty$ (Amit *et al.* 1985), allowing however the attractor state to have a large overlap with a number $s > 1$ of stored patterns.

The result is this: there is no retrieval above $T = 1$ (the system is paramagnetic there); immediately below that critical temperature only pure retrieval exists; however, as the temperature decreases, a growing number (at $T \to 0$: a number exponentially growing with p) of different mixtures of 3, 5, ... patterns become locally stable. Although the true retrieval states are the most stable down to $T = 0$ (they have the lowest free energy), the total basin of attraction of the spurious memories overweighs in a random search.

Mean-field theory is much of the truth about our model but not the full truth. It contains all kinds of averaging that can smear out eventual fluctuation-driven effects, and it is used in conjunction with the limit $N \to \infty$ that erases finite-size effects. In particular, fluctuational barriers of $\mathcal{O}(1)$ can seriously influence the dynamics of a finite computer model. That can be a vague way to interpret the simulations of AGS who found that for a size up to $N = 1000$ learned patterns remain slightly noisy attractors even for an overloading as high as $\alpha = 0.16$.

For larger sizes $N \geq 2000$ that disappears, however one finds something more interesting: in associative retrieval experiments *i.e.* starting relaxation from a pattern, dynamics is trapped by one of many attractors of overlap ≈ 0.3 with that pattern. This has however nothing to do with the pattern: it is rather a *remanence* of the initial state (whether a pattern or not), reflecting non-ergodicity of the system. All that is apparently beyond the reach of a straightforward mean-field treatment.

5 DESCENDANTS OF THE HOPFIELD MODEL

The Hopfield model, by its simplicity and the possibility it offers for an in-depth theoretical analysis, became a point of reference for numerous studies. In this chapter we review the first generation of these developments. Their distinguishing feature is the way they treat the learning problem: they use a closed formula for the synaptic modification on storing a new pattern. The second generation, using slower but more powerful iterative learning algorithms, will be treated in Chapter 6.

5.1 The Hopfield model is robust against...

Sentences beginning like this abound in the literature. We know already one property in place under this heading: full connectivity makes the model robust against "thermal" noise. Indeed, if we measure the learning amplitude λ in "biological" units, independent of the number N of neurons, then the critical temperature characterizing the noise level that would destroy retrieval turns out to be of $\mathcal{O}(\lambda N)$ (cf. Section 4.3). On the contrary, if each neuron is connected to $C \ll N$ others only (section 7.1), the critical noise level is reduced to $\mathcal{O}(\lambda C)$. The case of low activity (see below in Section 5.3) may deserve special attention: even for high connectivity the signal level is low which may reduce tolerance to thermal noise.

After people started to get acquainted with some of the more unexpected features of the model, perhaps thinking that these are probably due to its over-simplifications, it turned out quite soon that the results are extremely resistant to rather evil-minded changes one might conceive. Some of these astonishing properties, which became a principle attraction of the field, could be studied by means of mean-field theory (Sompolinsky 1986, 1987; Feigelman and Ioffe 1987). A short list follows here.

Dilution of active synapses causes a roughly proportional reduction of the storage capacity but very little increase of noise before overloading, even if as much as 40% of the synapses are randomly cut. This is reassuring in view of the gradual drop in the number of synapses during our life. Moreover, this is a most important feature listed under the heading of *fault-tolerance*, a major claim in favour of artificial neural nets.

Static synaptic noise, *i.e.* Hebbian learning done on a random background instead of tabula rasa, has similar effects, with good memory action up to a quite high noise level.

Nonlinear learning algorithms are unavoidable in any real application, biological or artificial. The simplest model version, studied in the Sompolinsky papers, is to evaluate the Hebb-rule formula (4.1) but take a nonlinear function of the result as the synaptic strength. In particular, *clipping* the synaptic strengths to just $+1$ or -1 according to the sign of (4.1) (eventually adding 0 as a third possible value for "don't know" if the Hebb-rule result is very small) is quite attractive for computational implementations. Again, although one might expect much worse, the performance is equivalent to that of a tollerably diluted network. The theory of nonlinear synapses is not simple; alternative treatments exist as well (van Hemmen 1986).

A slightly different kind of nonlinearity, originally motivated by the biological requirement of leaving the sign of the synaptic strength unchanged on learning, is to multiply or divide each synaptic strength by a fixed positive number, according to which of the two leads to a change in the Hebbian direction. This "multiplicative learning" (Geszti 1986, Németh and Geszti 1988) results in a storage capacity depending on the size of the system, eventually reaching a maximum at some optimal size of the system.

Multiple synapses, the output of which depends on the simultaneous inputs from several neurons (as modelled most simply by a product of input activities), improve the storage capacity of the network but it is more difficult to develop efficient learning algorithms to them, although many results are available (Peretto and Niez 1986a, Personnaz *et al.* 1987, Baldi and Venkatesh 1987, Psaltis

et al. 1988, Horn and Usher 1988). Their biological reality seems to be supported by anatomic observations but their physiology is not quite clear.

Asymmetric synaptic strengths violate the principal theoretical claim of the Hopfield model: the existence of a monotonously changing energy-like function. Nevertheless, with a considerable amount of asymmetry the associative memory action still remains; moreover – a considerable advantage for potential applications – the retrieval is often much faster than for the symmetric case (Hertz *et al.* 1987, Kinzel 1987). This is comforting, since symmetry is biologically very unrealistic indeed (in particular, synapses off a given neuron are either excitatory or inhibitory; with the requirement of symmetry, this would make these two populations of neurons unconnected to each other: cf. Shinomoto 1987). A sharper statement is that what one really needs is not the permanent attractors forced on us by the symmetry of too simple models, only temporary halts to mark meaningful things (Parisi 1986b; cf. Chapter 1).

In the same paper of Parisi (1986b) another shortcoming of the Hopfield model is pointed out, asymmetry being mentioned as a possible remedy. The network, if started from an arbitrary initial configuration, will relax *somewhere*. If you started close to one of the learned patterns and the network is not overloaded, the final configuration will be close to the initial one. That can happen however – even if this is not very probable – if the initial configuration is far from all learned patterns: just by observing the Hopfield net, there is no way to decide whether a successful retrieval happened or not. Fontanari and Köberle (1988a) took a short cut to a solution, putting the desired feature into the model by hand. They constructed a model in which – in a rather unusual way – connection strengths depend on the actual configuration, in such a way as to detect overlaps with the stored patterns, and whenever there is no large overlap with any of them, switch on large negative diagonal couplings. These then, under *parallel updating dynamics* (another deviation from the Hopfield reference, instead of asymmetry), force all spins to flip cyclically to and fro, so to say, ringing the bell to indicate that no retrieval happened. The desired goal is achieved by far more economically in a low-activity network with inhibitory bias (Buhmann *et al.* 1989): there the

no-retrieval configurations approach just zero activity. Other ideas related to the same problem can be found in Shinomoto (1987) and Skarda and Freeman (1986).

Later on it turned out that symmetry, which allowed the first rough insight into the dynamics of retrieval and provided us with the remarkable statistical-mechanical treatment of its statics, is rather a plague when one tries to understand finer dynamical details, since it is close to impossible to do the bookkeeping of signals just sent out and already coming back on the reverse synapse. In this respect fully asymmetric models recently became rather popular, as will be reviewed in Chapter 7.

What we mentioned so far was mostly random asymmetry. On the other hand, a carefully tailored asymmetry can be used to store and retrieve *sequences* of patterns. This is a big subject on its own, and will be described in more detail in Section 5.5.

Last but not least, an electronic network with continuous characteristics, impedances and the corresponding time lags in processing, has been demonstrated to successfully implement Hopfield-model-like action, under some well-defined conditions (sufficient gain, sufficiently fast switching: see Hopfield 1984).

5.2 The projection rule

Firing and non-firing neurons usually carry some specific information (like black and white pixels of an image), not just a random code for something abstract. In that case, as a rule, different patterns are not orthogonal, *i.e.* they have an overlap of $\mathcal{O}(1)$ instead of the random $\mathcal{O}(1/\sqrt{N})$. It is a major practical disadvantage of the Hopfield model that for non-orthogonal patterns it is no more applicable as an associative memory: to an input initial configuration equal to some of the patterns, it relaxes to something quite different. Apparently, the root of the troubles is that "random overlaps" with other patterns cannot be regarded as random noise any more in this case.

The task of errorless storage and retrieval of non-orthogonal patterns can be solved if one abandons the simple Hebb rule.

Multi-step iterative learning rules (Chapter 6) can do the job to perfection. For many non-biological purposes however, both for physical modelling and artificial implementations, it is very advantageous that a non-iterative solution to the problem also exists. This is the so-called *projection rule*. First introduced by Kohonen and Ruohonen (1973) in a slightly different context, it is not quite fair to regard it as a descendant of the Hopfield model; for the present purpose it was rediscovered by Personnaz, Guyon and Dreyfus (1985, 1986). It is also mentioned as the *pseudoinverse method.**

The projection rule is a closed formula to calculate synaptic strengths from

$$J_{ij} = \frac{1}{N} \sum_{\mu,\nu} \xi_i^\mu (\mathbf{q}^{-1})_{\mu\nu} \xi_j^\nu, \tag{5.1}$$

where $\mathbf{q}^{-1}$ means the inverse of the matrix of overlaps between two patterns, whose element $q_{\mu\nu}$ is defined in Equation (4.6).

It is easy to check (see below) that this choice of connection strengths gives perfect, errorless recall of all patterns. The only necessary condition is that the inverse matrix $\mathbf{q}^{-1}$ should exist, for which a non-vanishing determinant of $\mathbf{q}$ is needed, therefore linearly independent patterns. Their maximal number is N for N neurons, therefore the storage capacity of the projection rule is $p_{max}/N = \alpha_{max} = 1$. However, teaching the network close to saturation may reduce the attraction basins of the patterns close to nothing.

The proof of errorless recall for all patterns is quite trivial. What should be checked is the positivity of Δ_i^ν (see Equation (4.7)). Indeed, with (4.6) and (5.1) more can be proved:

$$\sum_j J_{ij} \xi_j^\lambda = \sum_{\mu\nu} \xi_i^\mu (\mathbf{q}^{-1})_{\mu\nu} q_{\nu\lambda} = \xi_i^\lambda \tag{5.2}$$

* If $\xi_i^\nu \equiv (\xi)_{i\nu}$ is regarded as a matrix in its two indices then the overlap matrix is $\mathbf{q} = \xi^T \xi$ where the upper index T means the transposed of a matrix, and the combination $\sum_\nu (\mathbf{q}^{-1})_{\mu\nu} \xi_j^\nu$ in equation (5.1) can be recognized as the so-called Moore-Penrose pseudoinverse of this matrix. Powerful methods exist for the calculation of pseudoinverses (see e.g. Albert 1972).

which implies $\xi_i^\lambda \sum_j J_{ij}\xi_j^\lambda = 1$ for all i and λ, without any noise. (5.2) means that each pattern is an eigenvector of the connection matrix J_{ij} with eigenvalue 1. In other words, in the configuration space J_{ij} is a projection matrix onto the subspace spanned by the learned patterns. That is why this is called the projection rule.

As it will be discussed in detail in section 6.4, the projection rule is just one of a continuum of possible solutions to the problem of errorless storage. Indeed, any set of positive eigenvalues instead of the uniform 1 would do the job; moreover, patterns need not be eigenvectors at all. The choice is motivated mainly by the handiness of the resulting model.

The advantages of the projection rule over Hebb's rule are considerable: robustness against overlap between the patterns, errorless recall, much larger storage capacity. Even the theoretical tractability remains (Kanter and Sompolinsky 1987a). The price to be payed for the advantages, in the present formulation, is the loss of *locality:* calculation of the inverse matrix $\mathbf{q}^{-1}$ requires information about all components of all patterns, *i.e.* the firing or non-firing of neurons not connected with the given synapse. This would make the rule biologically quite unplausible, although still useful for computational applications.

Locality comes back however, if one gives up having a closed formula and accepts an iterative algorithm for learning a set of patterns; in particular, the so-called *Adaline rule* converges just to the projection-rule result (*cf.* section 6.2).

5.3 Low activity and biased networks

Even if encoding quite abstract things, firing and non-firing are not equivalent for the hardware of either the brain or any kind of electronics. In particular, neurons in a brain usually fire at a quite low rate; most of the time they are quiescent. This means that the network carries out its tasks – including memory recall – at low values of the activity a measuring the mean fraction of neurons in the firing state. Since $a = (m + 1)/2$, now $m = \langle S_i \rangle$ is biased close to -1.

Low firing rates may seriously question the whole framework of the frequency code we have built everything upon. We return to this question at the end of this

Section. First, however, just take things at their face and use the formalism for low activity, which is put into the model as a parameter.

If we impose a common low activity, i.e. a common negative value of m as a "bias" on the learned patterns as well, then these patterns are no more orthogonal: if their components are independent and random apart from the fixed mean spin m, then the mean overlap of any two different patterns is m^2. It would be too sloppy to call such patterns "correlated"; they are just non-orthogonal.

A brute force way would be to use the projection rule. However, more *ad hoc* methods lead faster to the point. It turns out (Amit *et al.* 1987) that if the learning rule is modified from (3.1) to

$$J_{ij} = \frac{1}{N} \sum_{\mu} (\xi_i^\mu - a)(\xi_j^\mu - a), \tag{5.3}$$

that restores a zero average for the synaptic noise; if in addition the retrieval dynamics is also restricted to the subset of configurations biased to activity a, the associative memory works well. The storage capacity is even found to increase for large bias.

That is a point where one should remember that what we call "storage capacity" is a rather arbitrary thing. For everyday use a more suggestive quantity is the *information content* of the patterns, as measured by the Shannon entropy associated with the freedom of choosing p stored patterns in many different ways from the allowed set.

For the original Hopfield model each pattern component can be $+1$ or -1 with equal probability: that means 2^N equally probable possibilities and a corresponding information content of $N \ln 2$ (or N bits) per pattern. Neglecting for simplicity the information loss due to retrieval errors (which is not difficult to take into account), for the storage of αN patterns on a fully connected network this is αN^2 bits, *i.e.* $\alpha \leq 0.14$ bits of information stored on each of the N^2 continuously teachable synapses. For comparison, even a binary ("clipped") synapse would store just one bit if used in an isolated (non-associative) mode of operation. This is the price to pay for associativity, robustness etc.

Now biasing restricts the available patterns, and the amount of information stored per pattern is reduced to $\ln \binom{N}{Na} \approx -N[a \ln a + (1-a)\ln(1-a)]$. This, in the calculation of Amit *et al.* (1987), brings the information contained in all stored patterns monotonously down to zero for $a \to 0$.

Later, as catalyzed by Gardner's famous work (see below in Section 6.3), it turned out that with slight modifications (Tsodyks and Feigelman 1988, Tsodyks 1988) the storage capacity – as measured in the number of stored patterns – can be made to even *diverge* with bias, and the information content decreases only slightly, not to zero.

To achieve this, a tunable threshold should be introduced for retrieval. Moreover, it is advantageous to turn to V-variables: 1 for firing, 0 for quiescent. Then, for patterns $\eta_i^\mu = 1$ or 0, (5.3) would be modified to

$$J_{ij} = \frac{1}{N} \sum_\mu (\eta_i^\mu - a)(\eta_j^\mu - a). \tag{5.4}$$

For $a \ll 1$ this implies many weak synapses and a few strong of them. That reminds an old and long forgotten predecessor: the *Willshaw model* (Willshaw *et al.* 1969) that in these days enjoys a great return. In this model of associative memory one uses synaptic strengths clipped to 0 or 1; $J_{ij} = 1$ ("occupied synapse") if in any of the stored patterns both i and j are firing, 0 ("empty synapse") otherwise.

The Willshaw model with low-activity patterns – in the present context this is called *sparse coding* – has an information storing capacity much higher than the plain Hopfield model. To see this, following Nadal and Toulouse (1989), let us calculate the probability q that a given synapse is occupied ($J_{ij} = 1$) after learning p independent patterns of activity ("coding density") a. For a single pattern this probability is a^2; that of an empty synapse is $1 - a^2$; the same for p patterns is $(1-a^2)^p$; finally, $q = 1 - (1-a^2)^p$. With $a \ll 1$, this gives

$$\ln(1-q) \approx -pa^2. \tag{5.5}$$

We need this q to calculate the probability of erroneous retrieval. In the

Willshaw model the retrieval of a pattern is carried out by setting the neuronal threshold as high as $M = aN$: if the network is in a stored pattern configuration, the inputs to all those neurons which are firing in the pattern are granted to reach this value. Error can occur only on a neuron that ought to be *non-firing*, if it happens to receive the same input: for that, M given synapses must be occupied by chance. The probability of this is q^M.

Since the fraction of those bits in which an error can occur is $1 - a$ and that of those where it cannot is a, the noise/signal ratio is $((1-a)/a)q^M \approx q^M/a$ (for $a \ll 1$). Since M is a large number, this depends sensitively on q, and the limit of errorless retrieval, noise/signal≈ 1, defines rather sharply the limiting

$$q = a^{(1/M)} \tag{5.6}$$

allowing good retrieval.

And now the information capacity. If we are inside the errorless range, the information content of a sample is $\ln \binom{N}{M} \approx -N\big(a \ln a + (1-a)\ln(1-a)\big)$. For p patterns p times as much, which for $a \ll 1$ reduces to

$$I \ln 2 = \frac{N}{a}(-pf^2) \ln a. \tag{5.7}$$

(the factor of $\ln 2$ has been separated out to measure information in bits). Using (5.5) to (5.7) and $a = M/N$, one finally obtains for the boundary of the good retrieval region

$$I \ln 2 = N^2 \ln(1-q) \ln q. \tag{5.8}$$

This still depends on q and it is maximal if $q = \frac{1}{2}$. For this optimal case the information content is $N^2 \ln 2 \approx 0.69N^2$ bits: about five times as much as for the Hopfield model.

The Willshaw model and its continuous-synapse version (Nadal and Toulouse 1989) – both in the sparse coding limit – seem to be ideal in many respects. Their information content is very high without slow learning procedures, performing well even if overloaded (in this respect the continuous-synapse version is much more

tolerant, if one tunes the threshold to restore low activity in spite of the growing excitatory synaptic strengths): they can keep a large amount of information in the form of many erroneously recalled patterns. This makes sparse coding a promising direction for future research.

Let us return now to the serious biological problem posed by the observed low activities. According to the original Hopfield picture, in the state of retrieving a pattern a few neurons would fire for some time hundreds of spikes per second (limited only by the refractory time of the neuron). Instead, spike frequencies of any neuron are seldom higher than a few tens. How, then, is information encoded into firing patterns at all?

Treves and Amit (1989) offer a way out of the dilemma: in their model a low average activity network retrieves a pattern if the active neurons of that pattern fire during some time *one after the other* in a random sequence, providing a relatively large overlap with the pattern *on the time average*. This is like making a chord sound for some time on the piano by performing it *broken* (in "arpeggio"), which is the natural way to do it on a percussion instrument emitting spikes, like our brain. In Treves and Amit's instrumentation the scenario is implemented on an active network of excitatory neurons (with excitatory outgoing synapses only) carrying the information; connected to a background of rapidly (for physicists: "adiabatically") responding inhibitory ones, providing just a nonlinear negative feedback to keep the excitatory activity on the desired low level.* Whether the random order the spikes play the patterns (à la "musique aléatoire") is unavoidable, or in some variants (*e.g.* with time lags and/or coincidence detection properties built into the model) neurons can fire in a fixed order that would make the whole thing more similar to *synfire chains* (Abeles 1982, cf. also Chapter 1) and perhaps connect the issue to that of time sequences (Section 5.5), remains to be seen. The time order of spikes in a pattern may carry extra information or not: it can be just be a hardware necessity, like breaking a chord on the piano in the handiest way.

* This may be a rational explanation for the "normalization of activity" often used in early models of neural network dynamics (for a review see Clark *et al.* 1985) as an arbitrary mathematical tool to avoid unrealistic runaway effects.

5.4 Hierarchical storage

Another interesting special case of non-orthogonal patterns is that of hierarchically ordered knowledge. This would mean that the most important rough features of some kind of objects are stored on a base level, then an increasing resolution of fine details appears on subsequent levels.

The varieties of fine details are grouped on each level according to the rough classification they are compatible with. This gives the suggestive picture of a branching tree, thinner branches corresponding to finer details, the leaves of the tree being the ultimately resolved items. Those leaves which can be connected through a closer branching point are closer related (figure 5.1.a).

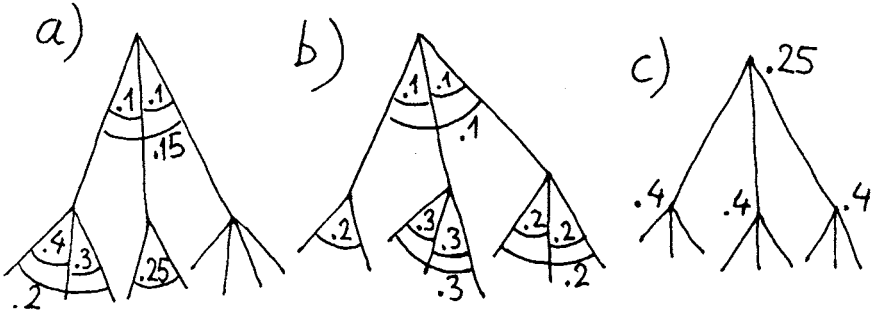


Figure 5.1 A hierarchical tree: a) not ultrametric b) ultrametric, c)ultrametric with levels of identical nodes. Numbers at nodes denote the overlaps between branches converging there.

Here again, the simplest task one can think about is to store hierarchically ordered patterns in the memory of a Hopfield-type network, then retrieve it associatively: starting from an input initial state within the basin of attraction. To make it really useful, solutions should allow to *choose the level* of retrieval, giving more ease if you are interested in rough features only. A slightly different aspect of the problem would be *restricted associativity*: if only the rough features of the

input state are within the basin of attraction of the common rough features of a class of patterns, one should be able to identify the class and not bother about unidentifiable details.

An engineer's first guess would be to assign different levels of the hierarchy to different parts of the network. Although such solutions to the problem exist indeed (see below), it is one of the little miracles of the Hopfield paradigm that hierarchical storage is possible on a homogeneous, fully connected network as well, just by an appropriately chosen set of connection strengths. By now this should be no big surprise to the Reader, since the projection rule (see above) or any of the dynamical learning rules (Chapter 6) have no difficulty with storing non-orthogonal patterns.

Some hierarchical sets have the special feature that any pair of patterns whose closest connection goes through the same (say, the l-th) node has the same overlap q_l (figure 5.1.b). In this case we have to do with a so-called *ultrametrically* ordered hierarchy, which simplifies the treatment considerably (Rammal *et al.* 1988). Ultrametrics implies (usually it is even defined so) that for any three patterns α, β and γ at least two of the three overlaps $q_{\alpha\beta}$, $q_{\beta\gamma}$ and $q_{\alpha\gamma}$ are equal, and the different one (if any) is the greater. This intriguing property is natural in its way: the greater overlap is between two patterns belonging to the same smaller bunch; the smaller overlaps are those of both with a more remote third. The non-trivial point is the equality of not less than two (actually three) overlaps if the three patterns are pairwise connected through the same node: this is the property we mentioned to introduce ultrametrics. A further simplification results if nodes can be arranged into levels, and nodes on each level are characterized by the same overlaps and multiplicities of branches converging there (figure 5.1.c).

The study of ultrametrically ordered patterns was triggered by the discovery of an ultrametrical hierarchy of free energy minima of the Sherrington-Kirkpatrick model of a spin glass (see Mézard *et al.* 1988). It was expected that if one starts from a spin-glass instead of tabula rasa, then hierarchically ordered patterns can be accomodated in some of the pre-existing free energy wells (Toulouse *et al.* 1986). This expectation was not satisfied: spin-glass ultrametricity is a very sensitive

flower that faints as soon as touched by something as crude as a stored pattern. (Patterns must be like that: otherwise the storage would not be robust, fault-tolerant etc.)

Nevertheless, turning back to the solid ground of tabula rasa, it is possible to generate an ultrametric set of patterns with given overlaps and devise a modification of Hebb's rule to store patterns from that set (Parga and Virasoro 1986, Feigelman and Ioffe 1987). More systematically, one can use the projection rule to teach an ultrametrical set of patterns. If, again, overlaps are defined beforehand and not deduced from the actual patterns, the task can be carried out analytically (Cortes *et al.* 1987). Moreover, the desired level-selective retrieval can be implemented by shifting the neuronal thresholds, in the quasi-spin language equivalent to turning on an effective external "magnetic field". (Krogh and Hertz 1988). The results seem not to be sensitive to the details of the storage rule used.

If one gives up the fully connected Hopfield architecture, a wide variety of inhomogeneous solutions to the hierarchical storage problem can be found. So far, the following proposals have been advanced:

a) One possibility (Dotsenko 1985) is to retain full connectivity, defining however a nested hierarchy of blocks of spins. The strongest bonds connect spins within the smallest blocks; then there are weaker and weaker bonds between spins belonging to different blocks on a bigger and bigger scale. For block-to-block bonds the Hebb rule is used through "block spins" defined iteratively as the *sign* of the sum of lower-order block spins within the bigger block. These block-to-block bonds are tuned to be weak enough to avoid *on the average** spoiling the retrieval of what is stored in the stronger intra-block bonds. In the Dotsenko model the rough features of the full patterns appear on the patterns of block spins. Therefore – somewhat paradoxically, and contrary to the homogeneous models discussed above – it is the weak bonds which contain the rough features easy to retrieve, whereas fine details are stored in strong bonds and are retrievable only by investing more

* It might be timely to reconsider the same model in the light of dynamical learning rules that probably allow one to defend the information stored on the small blocks without errors.

energy. Then tuning the level of retrieval might be implemented by changing the noise temperature.

b) A different possibility to form a network of nested clusters, advanced by Sutton *et al.* (1989) based on anatomic information about cortical columns, is to distinguish so-called *projection elements*: special-purpose neurons in a fully connected cluster to give an output signal (in a neurobiologist's language: to *project*) to another neuron belonging to a remote cluster. Sutton *et al.* formulate the task for such a network in a somewhat unusual manner, as the storage and retrieval of correlations between different details of different patterns; the model implements this task by asymmetric synapses and time delays in the "projection" neurons. However the idea of assigning block connections to special neurons may prove a useful architecture for dynamical models of the (for physicists) more conventional type. Thinking in terms of associative memory, here, instead of Dotsenko's block spins, it would be just the projection neurons that might carry the rough information.

c) An obvious alternative (as sketched by Dotsenko 1986) is a layered network, in which connections between subsequent layers of spins store subsequent levels of detail. In a recent interesting study Gutfreund (1988) describes an ingenious version in which retrieval dynamics is run layer by layer starting from the one storing the roughest features. Then completing the retrieval of a pattern of rough features provides a beneficial bias for the next layer, reducing its noise level when looking for finer details. This – depending on the number of layers – can increase the storage capacity enormously. Essentially the same idea is described by Sourlas (1988). The two versions – a predecessor of which appears already at Feigelman and Ioffe (1987) – differ in the way the first-layer-pattern acts on the second layer. In this construction both the choice of level and restricted associativity are trivially satisfied: one can just stop before the second layer might do anything.

d) In a very non-Hopfield architecture, a two-dimensional topologically ordered layer of neurons with strong close-neighbour bonds possesses a very natural way to do hierarchical storage for any kind of information with a topology (i.e. neighbourhood relations, like neighbouring directions of the visual field when stor-

ing pictures): fine details appear localized on one neuron or a few neighbouring ones, whereas gross features are represented as bigger areas containing the related individuals (Kohonen 1988). The formation of such a topologically ordered network will be treated in more detail in Chapter 10.

A real application demanding hierarchically ordered information storage and retrieval is expert systems. No breakthrough in this direction happened so far within the field of dynamical feedback neural nets.

5.5 Time sequences

If you want to sing a song or recite a poem, you need not retrieve the whole at a time (although the best professional performers nearly do so). Rather one line brings up the next. Modelling the storage and retrieval of similar sequences of patterns is one of the potentially most promising applications of dynamical feedback networks, since dynamics appears here in the definition of the task itself.

Already Hopfield (1982) suggested that connection strengths

$$J_{ij} = \frac{1}{N} \sum_{\mu} \xi_i^{\mu+1} \xi_j^{\mu} \tag{5.9}$$

would act in the direction of stepping from the μ-th pattern to the $\mu + 1$-th and so on, through the sequence. Indeed, then $\sum_j J_{ij} S_j$ drives strongly towards $\xi_i^{\mu+1}$, whenever the network is in the configuration $\{\xi^{\mu}\}$.

Of course, these connection strengths are not symmetric, in accordance with looking for a sequence-like attractor, not just a fixed point.

Unfortunately it turned out that the thing does not work in this simple way, at least not for random asynchronous (sequential) dynamics that destroys the coherence of the patterns very soon and the sequence gets completely lost. Adding to (5.9) a term of the usual form (4.1) to stabilize the patterns does not help: depending on its amplitude, it is either too weak and then it changes nothing, or too strong and then there is no transition from one pattern to the other.

The way out has been found by Kleinfeld (1986) and by Sompolinsky and Kanter (1986) (see also Peretto and Niez 1986). They suggested that before

stepping on to the $\mu + 1$-th pattern, one should wait long enough to let the μ-th pattern fully stabilize. That would require a *time delay* (say, τ) in the stepping process: spins should be updated according to

$$S_i(t+1) = \operatorname{sgn} \sum_j \left(J_{ij}^{(1)} S_j(t) + J_{ij}^{(2)} S_j(t-\tau) \right), \tag{5.10}$$

where

$$J_{ij}^{(1)} = \frac{1}{N} \sum_\mu \xi_i^\mu \xi_j^\mu, \tag{5.11a}$$

$$J_{ij}^{(2)} = \frac{\lambda}{N} \sum_\mu \xi_i^{\mu+1} \xi_j^\mu \tag{5.11b}$$

and $\lambda > 1$. Suppose that the network is now in the pattern configuration $\{\xi^\mu\}$ but at time $t - \tau$ it was still in $\{\xi^{\mu-1}\}$: then both terms stabilize $\{\xi^\mu\}$. However, the time comes when τ seconds earlier the network was already in $\{\xi^\mu\}$: then the stronger (because of $\lambda > 1$) second term forces to flip into the new pattern $\{\xi^{\mu+1}\}$, and so on, stepping onto another pattern once every τ seconds (figure 5.2).

Associative retrieval in the present context means that if you start with a noisy (erroneous) version of the first pattern, the network gradually (after a few steps) begins to produce errorless patterns for the rest of the sequence. If patterns are arranged in a cycle, the repetition may be already completely free of errors.

It may have some advantages to work with a distribution of different time delays instead of a sharply defined τ; synaptic strengths can be clipped, etc. Various versions have been tried. With appropriate parameters you can even omit the "stabilizing" $J^{(1)}$ term. In other ranges of the parameters the desired sequential retrieval is replaced by chaotic behaviour (Riedel *et al.* 1988).

So far everything was simple: we succeeded to store and retrieve a single sequence of patterns, all different. Complications come now. How to treat a sequence with some of the patterns repeating (like the refrain of a song)?

To this end you have to keep a memory of a number of previous patterns: as many of them as needed to define the continuation unambiguously. The minimum

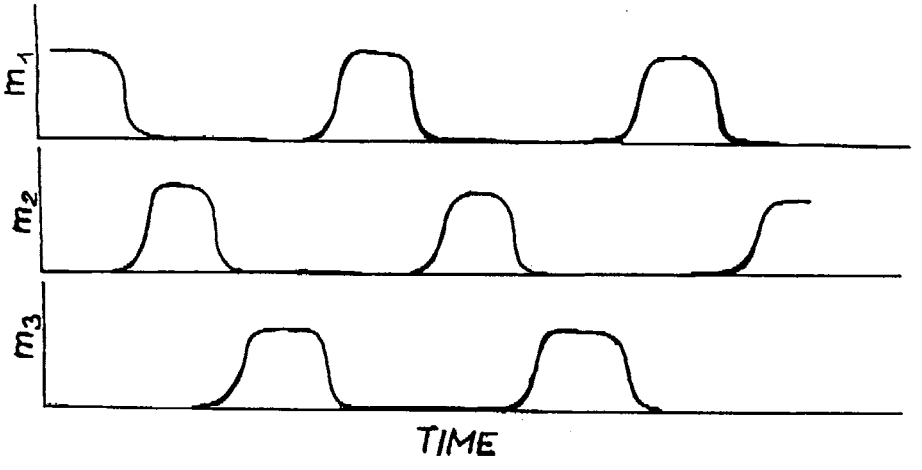


Figure 5.2 Time variation of overlaps of a network configuration with three patterns in sequence (schematic)

number needed is called the *order* of the sequence. A song with a single line as a refrain returning in each strophe (lines being the patterns of the sequence) is of order one: the line before the refrain defines the strophe.

Solutions to the task of storing such sequences have a wide variety of possibilities (see Guyon *et al.* (1988) where the network is studied under parallel dynamics, although part of the results can be used for sequential dynamics too; alternatives are described by Mori *et al.* (1989) and by Kühn *et al.* (1989)). For order-1 sequences a possible solution is to merge pattern ν and pattern $\nu - 1$ into a $2N$-dimensional vector, and use the pseudoinverse rule (Section 5.2) to calculate the synaptic strengths forcing the transition to the desired next pattern. More versatility is given by applying triadic synapses with partly time-delayed inputs, as earlier done to model bird song acquisition (Dehaene *et al.* 1987).

Although built-in time delays dominate the field of physical modelling for the moment, biologically (and for potential applications) they may be not the only mechanism to result in stepping from one pattern to another. Dynamical thresholds shifting in time so as to inhibit a neuron that has been firing for a long time (Horn and Usher 1989) offer a promising alternative. Noise can also have an active role in processing time sequences (Buhmann and Schulten 1987).

In the context of time sequences, associative retrieval is not the only thing to do with a neural network. If the stepping amplitude λ is tuned just a little bit too weak to excite transitions, a short input of external noise can force the network to step forward in a sequence of learned patterns: say, from one pattern saying "one" to the next saying "two" and so on: this is *counting* external signals unrelated to the patterns, *e.g.* chimes (Amit 1988).

Tuning λ still a bit weaker, a neutral (noise-like) external input is no more enough to push the network from one learned pattern to another, however if you present just *the next pattern* as an input for a moment, the transition takes place. With a sequence of inputs repeating that of the memorized patterns, the network carries out the corresponding sequence of transitions, thereby *recognizing* that the input is just what had been learned. At some bifurcation points the network can choose between two learned continuations, according to the actual input. All this requires noise – both thermal and synaptic – of a carefully adjusted level, mainly because of the danger of getting stuck in a spurious pattern composed of the active and passive partners of the forthcoming step (Gutfreund and Mézard 1988).

In artificial network applications – and perhaps in physiology too – an important option in time sequence processing is *chunking*: a large amount of information can be treated as a single pattern, or a short sequence of many-bit patterns, or a longer sequence of few-bit ones. Optimization with respect to network size, speed of processing, order of the resulting sequences that influences the network architecture needed etc., restricted by the important supplementary condition of some desired way of initializing of retrieval, may favorize solutions between, say, coding a poem as a sequence of letters, or words, or lines, or strophes, or as a single pattern (Guyon *et al.* 1988).

This may be a place to refer back to the problem of low activity (Section 5.3): a pattern displayed by asynchronous spikes *is* a time sequence of sub-patterns each consisting of a single neuron firing on a quiescent background, and it is possible that a living brain finds it advantageous to store it as a sequence of given temporal order. Then, even if this is a *pseudo-sequence* in which the temporal order carries no information, the order is there, and this can be a rational explanation for synfire chains (Abeles 1982).

5.6 Invariant pattern recognition

One of our basic abilities is to recognize objects, persons, landscapes once seen. We can do that in a way *invariant* to a very large set of transformations: shifting, rotating, colouring, enlarging or reducing the object, putting it into a different background, not to mention ageing of a person, may not prevent us from recognizing. To do the same by artificial computing is a problem of great practical importance, and developing computational tools for it became rather an industry (having its own journal called *Pattern Recognition*). A basic tool with amply demonstrated biological relevance is to form *feature detectors*, *i.e.* to evaluate characteristic features of the image, mathematically modelled as nonlinear functionals of the light intensity pattern, purposely chosen to enable one to identify categories of objects (Poggio and Reichardt 1976). Biologically this must be a highly complex function of the central nervous system; artificially it requires special insight to construct the appropriate functionals by hand. A physical analysis of the problem, largely independent of specific neural network implementations, has been recently given by Bialek and Zee (1987, 1988).

It would seem pretty natural to try modelling pattern recognition by associative memory: an image we see at this moment is not quite what we used to see long ago, however it may be within the basin of attraction of the old pattern, which can suffice for associating our present experience to it.

Unfortunately on closer inspection the problem appears a lot more difficult. At least for Hopfield-type networks, a pattern and its distorted, enlarged or

reduced, displaced, rotated etc. version are just orthogonal to each other, nothing like belonging to a common basin of attraction.

One might think of some clever modification of the learning rule, so as to line the basin of attraction with the potentially desired variants of each pattern. Such a learning rule has not been constructed so far, and it seems even improbable that looking for it would be the right philosopy: if not done in some unexpectedly natural way, the very demanding requirements would probably occupy an unreasonably large amount of memory.

Recently, an opposite approach begins to take shape (von der Malsburg and Bienenstock 1987, Kree and Zippelius 1988, Dotsenko 1988), consisting of essentially three stages. It leaves the patterns stored in the usual way, however generates a distorted, displaced etc. variety for the *input* pattern (possibly stored on an "input layer"): that takes fast operational memory only. Then the possible modifications of the input are scanned over the stored patterns, until eventually a large overlap is detected with some of them. The "attention" of the network being thus focused on that pattern, scanning is stopped and the selected pattern is accurately retrieved by the usual Hopfield-type relaxation.

Von der Malsburg and Bienenstock (1987) and Kree and Zippelius (1988) carry out this program for the specific problems of recognizing topologically equivalent graphs (say, variously distorted triangles) drawn on a screen. In Kree and Zippelius' implementation the generation of equivalent versions of the input and the retrieval of the correct prototype are done in physically different parts of the network, which is probably the practical way to do it.

Dotsenko's version (1988) is expected to work on a single network. It is restricted to translation by a vector $\mathbf{a}$, rotation by a matrix Θ and stretching by a scale factor λ. Patterns are on a two-dimensional screen where the neuron index i is written as a quasi-continuous position vector $\mathbf{r}$; scanning the versions $S^0_{\lambda\Theta\mathbf{r}+\mathbf{a}}$ of the input $S^0_{\mathbf{r}}$ happens by a slow noisy relaxation process varying $\mathbf{a}$, Θ and λ. The right prototype pattern once found, switching to Hopfield-type retrieval can be controlled automatically by a dynamical threshold locked to the varyingly distorted initial state: this is easy to put into equations but does not seem so

easy to implement in hardware, biological or electronic. Applicability is restricted to patterns and (or?) initial configurations drawn of smooth patches (not sharp lines), in order to provide sufficiently wide basins of attraction for the preliminary search.

In all versions tuning the noise level is an important item: not only in the process of ultimate retrieval should one avoid being trapped in a spurious memory, but also in the preliminary scan selecting the most promising pattern.

Apparently much further work is needed to see whether this "scan the input" way is a promising alternative to more conventional approaches to invariant pattern recognition, either in understanding the operation of our brain or in applications.

5.7 Learning within bounds and short-term memory

The overloading catastrophe of the Hopfield model is traced back to the level of synaptic noise growing indefinitely with a growing number of stored patterns (Section 4.2). It occurred already to Hopfield (1982) that this can interfere with the biologically plausible boundedness of synaptic strengths: for a learning amplitude λ and a synaptic strength remaining between A and $-A$, the signal-to-noise ratio should not grow over $\sqrt{N}\lambda/A$ for any number of patterns, which may hinder the breakdown.

In order to have an effect at all, the bounds on the synaptic strengths must be comparable to the noise level at the overloading breakdown: if much larger, they do not influence the storage capacity; if much smaller, they just suppress any memory storage.

The effect is not trivial; what happens has been clarified independently by Parisi (1986a) and by Nadal *et al.* (1986). They introduced a model in which Hebb-rule synaptic modifications are carried out only if they do not violate the bound. Their learning algorithm, called "learning within bounds", is this:

$$\begin{aligned} z &= J_{ij}^{\text{old}} + \lambda \xi_i \xi_j; \\ J_{ij}^{\text{new}} &= \begin{cases} z & \text{if } |z| \leq A, \\ J_{ij}^{\text{old}} & \text{if } |z| > A. \end{cases} \end{aligned} \tag{5.12}$$

This works in the following way. On learning a new pattern, only a minor fraction of the synapses would hit the bound. Each such event is a little kick of noise to the storage of all patterns present. Each pattern can survive a number of such events by getting just slightly erroneous like in the Hopfield model, then breaks down. Thus freshly learned patterns can be retrieved without error, while old patterns get gradually erased from the memory.

For a memory operating this way, Toulouse coined the term *palimpsest* that means an old piece of parchment from which one (a monk in a poor monastery) erases the written text in order to write something new on it.

As already mentioned, to get palimpsest operation, the bound A should be neither too large nor too small. If the learning amplitude is scaled – this time – to $\lambda = 1/\sqrt{N}$, the maximum storage capacity is obtained for $A = 0.35$: then after learning a very long sequence of independent random patterns by (5.12), about the last $0.041N$ patterns can be retrieved.

The learning-within-bounds model is a nice example of a well-established effect of experimental psychology mimicked, if not explained, by a model of neural networks. The case at issue is *short-term memory*: our ability to recall just a few things during a short time, without remembering it a few hours or days later. This is of vital importance: *e.g.* you could not understand current speech without remembering the whole of a sentence until the end of it.

The famous effect in short-term memory is *forgetting by interference*: it has been demonstrated by witty experiments that stored items are cancelled from the short-term memory by learning something new, not by any process of weakening or fatigue of the synapses. This is just what you see in the model. The well-defined capacity with respect to freshly stored events is also something present experimentally: it is often quoted that one can keep in one's short-term memory about 7 different "items". A warning against too direct optimism is that those "items" can be rather complex things, not just "patterns": short-term memory, whatever its mechanism, is embedded into a complicated cognitive environment.

Learning within bounds is clearly irrelevant for long-term memory: there is no indication that our storage capacity might get exploited close to its physical

limits set by finite synaptic strengths. This is the case where bounds are too high to matter.

A few more details about the theory of the learning-within-bounds effect: the synaptic strengths resulting from (5.12) are complicated nonlinear functions of all stored patterns, however they can be shown (van Hemmen *et al.* 1988) to be dominated by the usual Hebb-rule-like bilinear terms, all the rest adding up to a Gaussian noise:

$$J_{ij} = \sum_{\nu} \lambda_\nu \xi_i^\nu \xi_j^\nu + D \cdot y, \tag{5.13}$$

where y is a Gaussian random variable of zero mean and unit dispersion,

$$D^2 = \frac{A^2}{3} - \sum_{\nu} \lambda_\nu^2, \tag{5.14}$$

and

$$\lambda_\nu = \overline{J_{ij} \xi_i^\nu \xi_j^\nu} \tag{5.15}$$

can be determined as the mean value of a quantity carrying out a random walk between A and $-A$. In a long learning sequence this quantity is uniformly distributed in the allowed interval *before* learning pattern $\{\xi^\nu\}$; after learning its distribution is shifted in the positive direction to give it a positive mean value; then learning new patterns, independent of $\{\xi^\nu\}$, causes a random walk that smooths out the distribution, reducing λ_ν gradually to zero. That drives the older ones among short-term memories to get forgotten.

The model with connection strengths (5.13) can be solved by the replica method (van Hemmen *et al.* 1988, Pázmándi 1988); the calculation with the assumption of replica symmetry reproduces the simulation results very accurately.

On the other hand, the model provides an interesting insight into the limitations of the simple signal-to-noise reasoning. The straightforward application of noise analysis would describe aging of a given memory through the decreasing signal λ_ν (see above) sinking below the constant noise level. Although this nice-looking argument qualitatively describes the effect, quantitatively it fails. The

dominant effect here is the presence of the stronger fresh memories: an old memory, before disappearing in noise, becomes *unstable* with respect to the freshest pattern that exerts an irresistible sucking action on the dynamic flow (Geszti and Pázmándi 1987, Pázmándi and Geszti 1989).

6 PATIENT LEARNING

6.1 Setting the aim

The simple Hebb rule (3.1), although qualitatively it acts in the right direction of stabilizing the new pattern, cannot be expected to do the full job: obviously the routine of using equally large modifications at each synapse is too rough to avoid doing much harm to the storage of other learned patterns.

The overloading breakdown of the Hopfield model – although referring to this simple learning rule – is a warning of general validity that not any set of patterns can be memorized on a given network. However, if the task of learning is solvable at all, it is expected to be solved by some close-control iterative algorithm: taking one pattern at a time, checking where big errors are and where small ones and eliminating them by as little modification as possible, then proceeding to another pattern and so on, until convergence. Indeed a number of such iterative algorithms exist, but before studying them, let us specify what we expect to achieve.

An obvious necessary and sufficient condition for a pattern $\{\xi\}$ to be a fixed point of the dynamics is that the stability Δ_i of the pattern, as defined by equation (4.7), should be positive for all neurons i.

This however does not guarantee that $\{\xi\}$ should be an attractor of the dynamics, since it is possible that from an input initial pattern differing just in one bit from $\{\xi\}$ the dynamics flows away to something very different.

A transparent way to create a finite basin of attraction around a pattern is to fix a *minimum stability* κ by the inequality

$$\Delta_i \geq \kappa > 0, \quad i = 1 \ldots N. \tag{6.1}$$

A neural network of connection strengths satisfying (6.1) for all memorized patterns with a common finite κ is called a *strict-stability network*. Let us remark that the projection rule (5.1), by its projection property (5.2), obviously furnishes finite stabilities.

Inequality (6.1) is a condition to be satisfied independently for each neuron i by its N incoming synaptic strengths J_{ij} $(j = 1 \ldots N)$, for a number of patterns $\{\xi^\mu\}$ $(\mu = 1 \ldots p)$ to be stored. No symmetry is required – nor is usually obtained from the learning algorithms to be treated – between J_{ij} and J_{ji}.

The normalization introduced in equation (4.7) is a convenient way to make the stability bound κ a useful characteristics, independent of both the size N of the system and the scale of the connection strengths J_{ij}. This is not the only way however. Another possibility is to omit the normalization denominator, setting just $\Delta_i^\mu = \sum_{j \neq i} J_{ij} \xi_i^\mu \xi_j^\mu$, however restrict connection strengths either individually, *e.g.* by a condition like

$$|J_{ij}| \leq \frac{1}{\sqrt{N}} \tag{6.2}$$

(that includes the option of clipping *i.e.* requiring strict equality: see Section 5.1), or – what received much attention in connection with Gardner's work (see Section 6.4) – globally, by a *spherical constraint*:

$$\sum_{j \neq i} J_{ij}^2 = 1. \tag{6.3}$$

The individual-synapse constraint (6.2) implies that for a strict-stability network satisfying (6.1) there is a simple lower bound for the size of the basin of attraction, *i.e.* for the number δ of erroneous input bits which cannot prevent the network to relax to the stored pattern (Krauth and Mézard 1988). Indeed, δ is bounded from below by the number of errors corrected *in a single step* of parallel updating. If (6.2) holds, flipping one bit may change $\sum_j J_{ij} S_j$ by no more than $2/\sqrt{N}$. Then δ such changes cannot eat up even the worst stability κ if

$$\delta < \frac{\sqrt{N}}{2} \kappa. \tag{6.4}$$

This supports the intuitive connection between the size of the basin of attraction and the stability.

An iteration conducted so as to achieve a clean-cut aim is like a teacher continuously correcting the errors of a pupil: in the present context this is called

supervised learning. This is actually a richer category: in Chapter 8 we shall see that it extends over hetero-association tasks, requiring to associate a well-defined output to each given input. In neural computation such tasks are usually solved by feed-forward networks.

Supervised learning with an individual control over each task is not the only way data can be given a useful processing. Less authoritative is to find a reasonable output in accordance with some global requirement (*e.g.* similarities of input patterns should be reflected in some way in the outputs), leaving enough flexibility to adapt the solution to the particular features of the input set. This is called *unsupervised learning*, or – more expressively – *self-organization* (Chapter 10).

Finally, the simple Hebb rule is too abrupt to be categorized either as supervised or unsupervised.

6.2 Iterative learning algorithms

It is easy to develop iteration procedures that are devised to reach a finite stability κ for a set of patterns that need not be orthogonal. Iterations should be carried out independently for each neuron i until inequality (6.1) is satisfied.

Two popular versions, originally developed for hetero-associative tasks (see Chapter 8), are the basis of all subsequent developments. The first of them (Rosenblatt 1961) is the

PERCEPTRON ALGORITHM:

- start from any (*e.g.* random) set of connection strengths;
- enter an input pattern $\{\xi^\nu\}$;
- for a given neuron i check if the condition

$$\xi_i^\nu \sum_j J_{ij}\xi_j^\nu > c \tag{6.5}$$

is satisfied; if not, change each input weight according to Hebb's rule:

$$J_{ij} \to J_{ij} + \lambda \xi_i^\nu \xi_j^\nu \quad (j = 1 \ldots N;\ j \neq i); \tag{6.6}$$

- repeat it all for each input pattern cyclically or in random sequence until convergence;
- do the same for each neuron.

Finally, with the connection strengths achieved, you can normalize like in equation (4.7) to read off the resulting stability bound $\kappa = c/\sqrt{\sum_j J_{ij}^2}$.

The result of Perceptron-rule learning is something like a weighted Hebb rule: Hebbian terms (3.1) corresponding to different patterns are added up with different weights, according to the number of times they had to be activized before convergence. If not for other reason, this is clearly in the right way at least for very non-orthogonal patterns: one of them learned, the other needs much less weight in order to get stored in the memory.

Modifications of the Perceptron algorithm are numerous (Bruce *et al.* 1987, Gardner 1988, Abbott and Kepler 1988). A very efficient variant, called the *MIN-OVER* ('minimum overlap') algorithm (Krauth and Mézard 1988; the algorithm is also known by their names) is to find in every step the pattern which is farthest from satisfying the inequality (6.1), and do the next correction (6.6) with that pattern. Quantitatively, the criterion is to choose the pattern $\{\xi^\mu\}$ for which $\sum_j J_{ij}\xi_i^\mu\xi_j^\mu$ is the smallest. If – for fixed i – both J_{ij} and $\xi_i^\mu\xi_j^\mu$ are regarded as vectors in the single index j then the above expression is the overlap (scalar product) of these two objects and you have to choose always the pattern of minimum overlap with the actual connection strengths: that is the origin of the name.

Krauth and Mézard (1988) introduced one more refinement: it is to run until convergence (if possible) with a given value of c, then take a larger c and repeat the procedure. After final convergence achieved (that used to appear about $c \approx 10$), the above-mentioned normalization of the connection strengths is taken to read off the final value of κ_{opt} that is now called the *optimal stability* for the given set of patterns. This optimal-stability modification of the Perceptron algorithm is optimal in the sense of saturating the storage capacity calculated by Gardner (1988): see section 6.4.

The second fundamental iterative approach (Widrow and Hoff 1960), called the

ADALINE ALGORITHM

(for 'Adaptive Linear Network'), is to aim at strict equality in (6.1) with equal stabilities $\Delta_i = 1$. Hebbian steps proportional to the distance from this aim are iterated:

$$J_{ij} \to J_{ij} + \lambda \xi_i^\mu \xi_j^\mu \sum_\mu (1 - \sum_l J_{il} \xi_i^\mu \xi_l^\mu). \tag{6.7}$$

The requirement of unit stability for each pattern does not determine the connection strengths unambiguously. Therefore it is a non-trivial statement that (6.7) converges to a well-defined solution: the projection rule (5.1). Indeed, (6.7) can be shown to reproduce the so-called Gauss-Seidel iteration for calculating the inverse of the overlap matrix $\mathbf{q}$ appearing in (5.1) (Diederich and Opper 1987).

In programming layered feed-forward networks (see Chapter 8) it proved extremely important that the Adaline rule can be reformulated as a gradient-descent method: (6.7) is just a displacement

$$J_{ij} \to J_{ij} - \lambda \frac{\partial W}{\partial J_{ij}} \tag{6.8}$$

descending along the steepest path to the next local minimum of the "cost function"

$$W\{J_{ij}\} = \frac{1}{2} \sum_\mu (1 - \sum_l J_{il} \xi_i^\mu \xi_l^\mu)^2. \tag{6.9}$$

The famous "back-propagation algorithm" (section 8.2) that offered the first practical tool for solving practical tasks by neural computers grew out of this formulation. A drawback is that the solution of the task is apparently the absolute minimum of the cost function $W = 0$, whereas gradient descent can get stuck in a spurious solution: a higher-lying local minimum of W. Similarly to the Hopfield model (section 4.1), addition of "thermal" noise – possibly under carefully chosen "simulated annealing" schedules (Kirkpatrick *et al.* 1983) – can help a lot. This is however *learning dynamics*, not retrieval.

Somewhat more remote relatives of these procedures are those which use the result of a test relaxation to correct rough Hebb-rule learning. In this very intuitive direction the most solid approach is that of Pöppel and Krey (1987) who start a relaxation run from a pattern $\{\xi^\nu\}$, arrive at a different configuration $\{x^\nu\}$, and then update the connection strengths by adding $\lambda(\xi_i^\nu \xi_j^\nu - x_i^\nu x_j^\nu)$. Surprisingly enough, one can do just the opposite as well: starting with the same test relaxation from a pattern and adding the result in the form $\lambda x_i^\nu x_j^\nu$ to the connection strengths, *i.e.* 're-learning' the result of the relaxation, seems to be also beneficial to the memory, as argued by Dotsenko and Feigelman (1989).

The most exotic is the oldest of this direction: to start a relaxation process from a *random* initial state and add the result with a small negative coefficient $\lambda < 0$, *i.e.* *'unlearning'* the effect of random retrieval (Hopfield *et al.* 1983; verbally suggested also by Crick and Mitchison 1983). The declared effect of unlearning is to erase the most strongly attracting spurious memories while doing no harm to the true ones. Crick and Mitchison (1983) conjectured that this is a possible function of dream sleep (see also Chapter 11). The negative terms in the Adaline rule (6.7) or in the Pöppel-Krey algorithm are occasionally also mentioned as unlearning, however the context is quite different.

Perceptron and Adaline learning are fascinating dynamical processes in the space of coupling strengths. Their study, partly directed towards feed-forward networks, shows many interesting phenomena, like phase-transition-like slowing down close to saturation (Hertz *et al.* 1989a,b; Opper 1987,1988; Peretto 1988b). The subject is recently reviewed by Kinzel and Opper (1989).

It is relatively easy to show (see below: section 6.3) that the Perceptron algorithm and its Krauth-Mézard modification (like several other variants appearing in the literature) converge in a finite number of steps *if* a solution to the posed system of inequalities ((6.5) for each pattern) exists at all. The much more difficult question about solvability of the system (6.1) depending on the desired stability and determining the ultimate storage capacity of the system is addressed in the section 6.4. No proof of convergence exists for the Adaline rule; indeed, as said above, sometimes it converges to the wrong solution.

6.3 The Perceptron Convergence Theorem

The statement that the Perceptron algorithm converges if a solution exists is called the Perceptron convergence theorem (Rosenblatt 1962). We present here the proof, based on the original idea but following Krauth and Mézard's presentation (1988).

For the sake of compactness let us notice once again that condition (6.5) has to be satisfied for each i separately, therefore in dealing with a given neuron i, we need not care about the index i explicitely. Let us then introduce the notations $\xi_i^\mu \xi_j^\mu \equiv \eta_j^\mu$ and $J_{ij} \equiv J_j$. Further let us regard these quantities as components of the respective N-dimensional vectors $\vec{\eta}_\mu$ and $\vec{J}$. Then the requirement (6.5) can be written in the form $\vec{\eta}_\mu \cdot \vec{J} > c$. We notice that in our present notation the often-used normalization factor is $\sqrt{\sum_j J_{ij}^2} = |\vec{J}|$, and $|\vec{\eta}_\mu| = \sqrt{N}$.

The theorem says that if a vector $\vec{J}^*$ of connection strengths satisfying the system of required inequalities

$$\vec{\eta}_\mu \cdot \vec{J}^* > c \tag{6.10}$$

exists at all, then a (usually different) solution $\vec{J}_M$ can be obtained in a *finite* sequence of $M = \sum_\mu M_\mu$ steps of Perceptron learning (6.6):

$$\vec{J} \to \vec{J} + \lambda \vec{\eta}_\mu. \tag{6.11}$$

In this sequence pattern $\vec{\eta}_\mu$ is added to $\vec{J}$ just M_μ times. Starting from *tabula rasa* $\vec{J}_0 = 0$ this results in

$$\vec{J}_M = \lambda \sum_\mu M_\mu \vec{\eta}_\mu. \tag{6.12}$$

The proof consists in constructing a lower and an upper bound for $|\vec{J}_M|^2$ and comparing them. The lower bound results from the Schwartz inequality:

$$\begin{aligned} |\vec{J}_M|^2 |\vec{J}^*|^2 \geq & (\vec{J}_M \cdot \vec{J}^*)^2 \\ & = (\lambda \sum_\mu M_\mu \vec{\eta}_\mu \cdot \vec{J}^*)^2 > (\lambda M c)^2 \end{aligned} \tag{6.13}$$

(the last inequality follows from (6.10)), or

$$|\vec{J}_M|^2 > \frac{\lambda^2 M^2 c^2}{|\vec{J}^*|^2}. \tag{6.14}$$

The upper bound is a consequence of the finiteness of the number M of learning steps and the boundedness of the amount $|J_M|^2$ can change in each step. Indeed, after $t < M$ learning steps the task is not solved yet, and whatever the pattern $\vec{\eta}_\mu$ taught in the $t+1$-th step, it is one for which

$$\vec{J}_t \cdot \vec{\eta}_\mu < c. \tag{6.15}$$

Then

$$\begin{aligned} |\vec{J}_{t+1}|^2 = |\vec{J}_t + \lambda\vec{\eta}_\mu|^2 = |\vec{J}_t|^2 + 2\lambda\vec{J}_t \cdot \vec{\eta}_\mu + \lambda^2 N \\ < |\vec{J}_t|^2 + 2\lambda c + \lambda^2 N. \end{aligned} \tag{6.16}$$

The last two terms are the upper bound of growth in one step; for M steps after *tabula rasa*, the result is

$$|\vec{J}_M|^2 < M\lambda(2c + \lambda N). \tag{6.17}$$

Comparing the two inequalities (6.14) and (6.17), one obtains finally

$$M < \frac{(2c + \lambda N)|\vec{J}^*|^2}{\lambda c^2}, \tag{6.18}$$

i.e. a finite number of learning steps to solve the tasks, as asserted by the theorem.

The Perceptron convergence theorem is of course just an existence theorem and does not say much about the actual number of steps required to solve a task. Actually the original Perceptron algorithm is quite slow. Considerably better are the performances of the MINOVER algorithm. The steps of the above derivation can be used as a guide in developing other fast alternatives (Abbott and Kepler 1988a).

6.4 The Gardner capacity of a network

A thoroughly new perspective of neural network theory was opened by the brilliant physicist *Elizabeth Gardner (1957-1988)* whose untimely death shocked the statistical physics community. She developed a method to calculate the storage capacity of a feedback dynamical network, independently of any particular learning algorithm. As a result, the performances of learning algorithms can now be evaluated on the basis of how closely they approach the Gardner bounds.

The starting point of Gardner's approach is to focus on the 'phase space of interactions', *i.e.* the space spanned by the connection strengths J_{ij} (all independent, no symmetry being assumed). In this space one calculates the volume within which a given set of p patterns can be given a finite stability, their Δ_i^μ-s as defined in (4.7) satisfying inequality (6.1) with a finite stability bound κ.

The volume within which the task is solved still depends on the actual patterns. Averaging over a distribution of patterns that can be independent or correlated, one obtains the volume as a function of a few parameters $V(p, N, \kappa)$.

As the load of the memory grows, V shrinks until it vanishes at a given critical $p_c(N, \kappa) = N\alpha_c(\kappa)$. This value defines then the capacity of the network.

The principal difficulty of carrying through this program is the large fluctuations of the task-solving volume as a function of the random patterns. Because of this, averaging over the distribution of patterns has to be done with much caution.

As a matter of fact, for errorless retrieval of a whole pattern of N bits, the roughly N^2-dimensional phase-space volume is the *product* of N subvolumes corresponding to the independent rows of the connection matrix J_{ij}. Therefore it is not V but $\ln V$ that is a sum of many weakly correlated terms, distributed in a roughly Gaussian fashion, with a small mean square deviation. Consequently, instead of taking just the average $\overline{V}$, it is $\overline{\ln V}$ that is a well-defined physical quantity. However, averaging the logarithm is usually difficult. Fortunately physicists have got a tool for that: the replica trick (see Appendix A). That is how actually Gardner solved the problem.

The vector notation developed in the preceding section helps one to assess the problem. In this language inequality (6.1) with the definition (4.7) can be written in the form

$$\vec{\eta}_\mu \cdot \vec{J} > \kappa |\vec{J}|. \tag{6.19}$$

Introducing the unit vectors $\hat{J} = \vec{J}/|\vec{J}|$ and $\hat{\eta}_\mu = \vec{\eta}_\mu/|\vec{\eta}_\mu| = \vec{\eta}_\mu/\sqrt{N}$, this can be rewritten as

$$\hat{\eta}_\mu \cdot \hat{J} > \frac{\kappa}{\sqrt{N}}. \tag{6.20}$$

For a given pattern this defines a calotte on the surface of the unit sphere of spherically normalized interactions $\hat{J}$. The Gardner "volume" is the intersection of the p calottes belonging to the p patterns; this intersection has to be averaged over a distribution of the patterns. The storage capacity is determined by the number p_c of patterns above which it has a vanishing probability to have a non-vanishing intersection. It is obvious that for strongly correlated patterns the calottes have more chance to intersect and the capacity is growing higher. As the correlation approaches complete identity of all patterns, the capacity should approach infinity since if you remember one of them, you know them all.

The actual calculation (Gardner 1988) is quite lengthy. Alternative derivations (Gardner and Derrida 1988, Mézard 1989) regard the problem as statistical mechanics of the interacting system of connection strengths, acquiring one unit of energy for each pattern violating inequality (6.1). This system is treated by Gardner and Derrida (1988) by replicas, and by Mézard (1989) by the cavity method which is a mean-field-like theory with explicite corrections for strong coherent fluctuations (Mézard *et al.* 1988). I confess having started the writing of this book with the hope of finding a simpler derivation of the final results before submitting the manuscript; unfortunately I failed. Let us conclude this chapter by a summary of the much-quoted results.

For unbiased patterns the critical storage capacity $\alpha_c = p_c/N$ is given by

$$\alpha_c(\kappa) = \left(\int_{-\kappa}^{\infty} Dt(t+\kappa)^2 \right)^{-1} \tag{6.21}$$

where the now familiar notation $Dt \equiv (dt/\sqrt{2\pi})e^{-t^2/2}$ has been used.

For correlated patterns, all of the same magnetisation $m = \langle S_i \rangle$ (cf. section 5.3), the result is

$$\alpha_c(m,\kappa) = \left[\frac{1+m}{2} \int_{(vm-\kappa)/\sqrt{1-m^2}}^{\infty} Dt \left(\frac{\kappa - vm}{\sqrt{1-m^2}} + t \right)^2 + \frac{1-m}{2} \int_{(-vm-\kappa)/\sqrt{1-m^2}}^{\infty} Dt \left(\frac{\kappa + vm}{\sqrt{1-m^2}} + t \right)^2 \right]^{-1} \tag{6.22}$$

where v is determined by the equation

$$\frac{1+m}{2} \int_{(vm-\kappa)/\sqrt{1-m^2}}^{\infty} Dt \left(\frac{\kappa - vm}{\sqrt{1-m^2}} + t \right) = \frac{1-m}{2} \int_{(-vm-\kappa)/\sqrt{1-m^2}}^{\infty} Dt \left(\frac{\kappa + vm}{\sqrt{1-m^2}} + t \right). \tag{6.23}$$

For $\kappa \to 0$ (6.21) gives $\alpha_c = 2$ *i.e.* $p_c = 2N$, which was earlier derived by Cover (1965) and Venkatesh (1986). This bound is fourteen times as much as the capacity of one-shot Hebb-rule learning; twice as much as that of the projection rule. Efficient iterative learning rules (Bruce *et al.* 1987, Gardner 1988, Abbott and Kepler 1988, Krauth and Mézard 1988, Pöppel and Krey 1987, Gardner *et al.* 1989) saturate the Gardner limit; their family is occasionally mentioned as the "Gardner universality class".

The case of correlated patterns also simplifies considerably for $\kappa \to 0$. For the particularly interesting case of very strong correlation, $|m| \to 1$ which includes the particularly important low-activity case $a = (m+1)/2 \ll 1$ (section 5.3), the storage capacity diverges as expected (see above):

$$\alpha_c \sim \frac{-1}{(1-|m|)ln(1-|m|)} \tag{6.24}$$

i.e. $\alpha_c \sim 1/2a|\ln a|$ for low activity (see also Tsodyks and Feigelman 1988, Buhmann *et al.* 1989). This however does not mean a diverging amount of high information: the many almost identical patterns say almost the same.

The case of clipped synapses $J_{ij} = \pm 1$ proved difficult for this type of analysis. The reason is that the replica symmetric approximation (Derrida and Gardner 1988) is not correct for this case. An ingenuous numerical study (Krauth and Opper 1989) gives the result $\alpha_c \approx 0.82$, reproduced very accurately in the replica scheme by one step of replica symmetry breaking (Krauth and Mézard 1989).

Memory storage capacity, which is the number of robust metastable states applicable to storage and reliable retrieval of memories, should not be confounded with the total number of metastable states. The latter, like in spin-glasses, grows exponentially, not linearly with the size of the system, even under drastic changes like asymmetric dilution (Treves and Amit 1988).

This multitude of metastable states however cannot be used for memory storage, because they cannot be programmed independently: they arise as an uncontrollable by-product of the linearly many memory states satisfying the stability requirements, as counted by Gardner. Incidentally, the exponentially many metastable states are strongly correlated in spin-glasses too, as displayed in the plateaus of the Parisi order parameter function.

Another aspect that makes it easier to accept the only linear storage capacity is fault-tolerance. Although the synaptic strength is a continuous variable (if not clipped), what can be used for sufficiently stable and fault-tolerant storage of information is hardly more than whether a given synapse is strong or weak, *i.e.* each synapse can store roughly one bit of memory, which results in a storage capacity growing only linearly with the size of the network.

7 DYNAMICS OF RETRIEVAL

A detailed knowledge of how fast the retrieval happens is of obvious importance for any application. Since associative retrieval is expected, one would also like to know something about the size and structure of the basins of attraction: how close to a pattern one has to start to retrieve it, how much the chance of retrieval depends on random noise *etc.*

For theory these questions are difficult; partial results are only beginning to accumulate. The present chapter is intended to review these developments.

A model neural network is a *dissipative* dynamical system: there is no conservation of energy that would force it to keep on moving indefinitely. If started from some initial configuration, it will typically relax to some attractor that may be – necessarily if the connection strengths are symmetric – a fixed point, or a limit cycle formed of a few configurations, or a more complicated attractor on which the motion continues chaotically. All this is influenced by the presence of learned patterns and the particular algorithm used for learning.

To define an objective hopefully accessible for quantitative analysis, an obvious choice is to restrict one's attention to motion close to an attractor or, say, two attractors, by starting from an initial state having a macroscopic ($\mathcal{O}(1)$) overlap with one or two patterns and a microscopic overlap with all the rest. Then the simplest quantity to characterize the approach to (or divergence from) one of the selected patterns is their time-dependent averaged overlap with the actual configuration $S_i(t)$ at time t:

$$m_\mu(t) = \frac{1}{N} \sum_i \langle\langle S_i(t) \rangle\rangle \xi_i^\mu . \tag{7.1}$$

A less self-evident quantity of deep physical content is the overlap between two different configurations both having a macroscopic overlap with the same pattern (or two patterns) only, otherwise initially independent and getting correlated later on because their motion is guided by the same learned (quenched-in) patterns (Derrida *et al.* 1987):

$$q(t) = \frac{1}{N} \sum_i \langle\langle S_i(t) \overline{S}_i(t) \rangle\rangle. \tag{7.2}$$

This quantity characterizes the ill-definedness of what at first sight appears as an attractor: even if m_μ would come to a rest to signal that the system is trapped forever, $q(t)$ does not approach 1. This is a warning that averaged out in the mean overlap, there are a number of different configurations either belonging to an extended attractor or a cloud of attractors, as already conjectured on the basis of numerical evidence in the AGS paper: Amit *et al.* 1987.

The average $\langle\langle \ldots \rangle\rangle$ is *threefold*, which is the source of all our headaches. First of all, it contains averaging over the noisy thermal history of whatever may happen in all steps form 0 to t according to equation (2.7) which is reproduced here in a slightly paraphrased form (and restricted to zero external fields)

$$S_i(t+1) = \pm \xi_i^\mu \text{ with probability } \frac{1}{1 + \exp \mp 2\beta \left(\sum_{j=1}^N J_{ij} S_j(t) \xi_i^\mu \right)}. \tag{7.3}$$

Then follows an averaging over a distribution of stored patterns. Finally, one may wish to average over a distribution of different initial configurations. On applying (7.3) it should be specified whether updating is done in parallel for all spins or sequentially, for one (randomly chosen) spin at a time. Finally, there is the averaging over spins, indicated explicitely in (7.1) and (7.2).

It is very easy to calculate what happens in the first time step: it is just a slight extension of the signal-to-noise analysis described in Section 4.2 that gives this information. However as soon as we try to follow the process for more time steps, the motion becomes very complicated; mainly because the distribution of configurations generated by the random process (7.3) becomes more and more entangled on every time step, and the simple quantities (7.1) and (7.2), which are only a few projections of this distribution, do not contain enough information to determine their own values for later times.

7.1 The asymmetrically diluted model

It was a real breakthrough when Derrida, Gardner and Zippelius (1987, hereafter referred to as DGZ) defined a modification of the Hopfield model in which the dynamics becomes simple and – as theoretical physicists call it – *solvable*. This means that once the model with its own inherent simplifications has been defined, the desired quantities can be calculated exactly, without further approximations. In particular: this is a model in which the information contained in the overlaps $m_\mu(t)$ is sufficient to determine the same quantities for subsequent times. This is obtained in the form of a closed one-step iteration formula giving $m_\mu(t+1)$ in terms of $m_\mu(t)$ (see below: equation (7.10)).

The DGZ model is obtained from the Hopfield model by strong *asymmetric dilution*: incoming synapses J_{ij} and outgoing ones J_{ji} are regarded as independent entities, and most of them are cut at random, to leave only very few of them. Then two neurons, if connected at all, are almost always connected only in one direction: the model is *fully asymmetric*. Later it turned out (see below) that this property – even without the high dilution – makes the dynamics of the model a good deal simpler.

Here however let us exploit dilution as it stands, since it allows a simpler derivation. The main point is that if each neuron remains connected to $C \ll N$ other neurons only, and two-way connections $J_{ij} = J_{ji}$ are excluded, then the signals received by a neuron on different synapses carry information with very high probability from different connected chains of other neurons, with no neuron in common between two chains. Therefore the incoming signals are statistically independent from each other and the straightforward signal-to-noise analysis can be applied to calculate their added effect.

It should be born in mind that the dilution happens in a way independent of the length of the synaptic connections, therefore the remaining connections are still infinitely long-ranged.

To obtain useful results (see below: after Equation (7.9)), it is required that the number of remaining connections per neuron should be still large: $C \gg 1$.

According to a more careful estimate (quoted in the DGZ paper) the condition for this statistical independence is $C \ll \ln N$. Inverting the above inequality to $N \gg \exp(C)$ it turns out that $C \gg 1$ entails an enormously large N, which excludes the model as a biologically realistic representation of some part of the brain.

And now let us see the derivation.* Our aim is to determine $m_\mu(t+1)$ from the knowledge of $m_\mu(t)$ if possible.

Instead of fully specifying the initial configuration, we fix only the initial macroscopic overlaps $m_\mu(t = 0)$; then we will have to average over all initial configurations compatible with these overlaps.

As a direct consequence of the definition of the overlap,

$$S_i(t) = \pm\xi_i^\mu \text{ with probability } p_\mu^\pm(t) = \frac{1}{2}(1 \pm m_\mu(t)). \tag{7.4}$$

We apply parallel dynamics (this is simpler; for the diluted model the results do not depend too strongly on this option). Then from (7.3) for a given set of stored patterns we obtain

$$\langle S_i(t+1)\xi_i^\mu \rangle = \tanh\Big(\beta \sum_{j=1}^{C} J_{ij} S_j(t) \xi_i^\mu\Big) \tag{7.5}$$

that is still not averaged over the patterns (this is indicated by the single angular brackets), nor over spins: this is not quite $m_\mu(t+1)$ yet.

Let us substitute the connection strengths from (5.13) which is sufficiently general for all we care in this presentation (however limited to statistically independent, unbiased patterns):

---

* We follow a predecessor of the DGZ paper: Kinzel (1984) who derived the main results as an *ad hoc* approximation for the fully connected model; the role of asymmetric dilution has been clarified in the DGZ paper, along with the trivial generalization to finite temperatures and the less trivial extension to a macroscopic overlap with several patterns.

$$\langle S_i(t+1)\xi_i^\mu\rangle = \tanh\Big(\beta\big[\sum_{j=1}^{C}\lambda_\mu\xi_j^\mu S_j(t) + \sum_{j=1}^{C}\sum_{\nu\neq\mu}\lambda_\nu\xi_i^\mu\xi_i^\nu\xi_j^\nu S_j(t) + \sum_{j=1}^{C}\beta D y S_j(t)\big]\Big). \tag{7.6}$$

Let us first investigate the case when the configuration $\{S(t)\}$ has an $\mathcal{O}(1)$ overlap with a single pattern $\{\xi^\mu\}$. Then in the square brackets, except for the first sum, we see only noise terms independent of each other (because of strong asymmetric dilution, as discussed at the beginning of this section):

$$\langle S_i(t+1)\xi_i^\mu\rangle = \tanh\big(\beta\big[\sum_{j=1}^{C}\lambda_\mu\xi_j^\mu S_j(t) + \mathcal{D}\cdot y\big]\big), \tag{7.7}$$

where

$$\mathcal{D} = \Big(C\big(\sum_\nu \lambda_\nu^2 + D^2\big)\Big)^{1/2}. \tag{7.8}$$

Summation ought to be restricted to $\nu \neq \mu$. Since however the sum has $\mathcal{O}(C)$ terms of equal magnitude and $C \gg 1$ (this is the place to use this condition), retaining or excluding a single term would not change the sum.

The method was applied by Derrida and Nadal (1987) to learning within bounds (Section 5.7); in that case because of (5.14) one has

$$\mathcal{D} = \sqrt{\frac{C}{3}}A. \tag{7.9}$$

Let us remark that in the diluted model λ_μ has to be normalized not to N^{-1} but to C^{-1} (or in the learning within bounds case: to $C^{-1/2}$). Anyway, in the square bracket of (7.7) one can neglect the dispersion of the first sum which is of the order of $\sqrt{C\lambda_\mu^2}$, whereas the Gaussian term is $\mathcal{O}(\sqrt{C})$ which is much larger. Therefore the first sum can be replaced by its mean value $C\lambda_\mu m_\mu(t)$. Finally, to arrive at (7.1), the "self-averaging" $\frac{1}{N}\sum_i$ (see Appendix A) has to be done. For the low-connectivity case this is equivalent to averaging over random patterns, which

in the present case (macroscopic overlap with only one pattern) means averaging over the Gaussian noise y. The result is this:

$$m_\mu(t+1) = \int \frac{dy}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} \tanh\big(\beta[C\lambda_\mu m_\mu(t) + \mathcal{D}\cdot y]\big). \qquad (7.10)$$

On iterating (7.10) one can follow the dynamics of the retrieval process in the asymmetrically diluted model: the pattern $\{\xi^\mu\}$ can be retrieved if the overlap $m_\mu(t)$ converges with growing t to a non-zero attractive fixed point (figure 7.1). At too high noise level (small β: "thermal" noise; too big $\mathcal{D}$: synaptic noise caused by the other memories) or too small value of the signal amplitude λ_μ (advanced forgetting) the only attractive fixed point is $m_\mu = 0$; then retrieval is impossible.

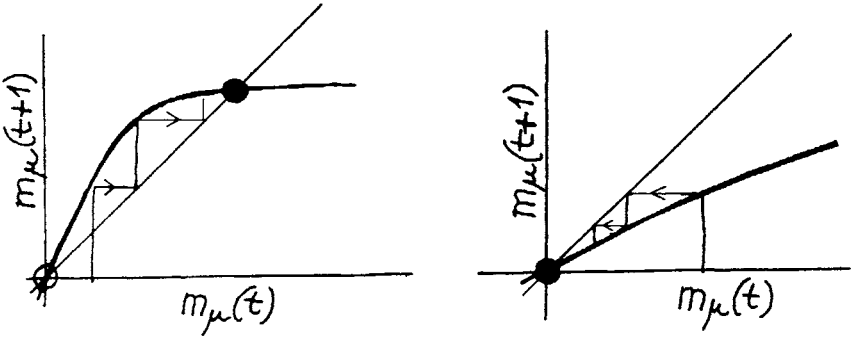


Figure 7.1 Iteration of Equation (7.10): a) with little noise to retrieval; b) with much noise to no retrieval

The transition between retrieval and non-retrieval corresponds in the present case to the Hopfield breakdown. In the asymmetrically diluted case however this happens by a continuous (second-order) phase transition: as noise grows, the overlap corresponding to the retrieval fixed point continuously decreases until it reaches zero at some critical noise level, which defines the storage capacity. For the asymmetrically diluted Hopfield model without any more complication, at $T = 0$, this happens if the number of stored patterns reaches $p_c = (2/\pi)C \approx 0.6366C$, being proportional to the connectivity, not to the number of neurons.

Generalization to the case when the spin configuration has a macroscopic overlap with several patterns, say $\{\xi^\mu\}$ and $\{\xi^\kappa\}$, is not quite trivial. Then on the right-hand side of (7.6) in the second sum there is a term $\xi_i^\mu \xi_i^\kappa C \lambda_\kappa m_\kappa(t)$ in which the factor $\xi_i^\mu \xi_i^\kappa$ can be randomly ± 1, with equal probabilities*. Now the averaging over patterns – that stands for self-averaging – is not done automatically by the Gaussian integral; one has to average over these two possibilities as well:

$$\begin{aligned} m_\mu(t+1) = {} & 0.5F\big(\lambda_\mu m_\mu(t) + \lambda_\kappa m_\kappa(t)\big) \\ & + 0.5F\big(\lambda_\mu m_\mu(t) - \lambda_\kappa m_\kappa(t)\big), \end{aligned} \tag{7.11}$$

where

$$F(x) = \int \frac{dy}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} \tanh\big(\beta\big[Cx + \mathcal{D}\cdot y\big]\big). \tag{7.12}$$

If we write down the analogous equation for $m_\kappa(t+1)$, we obtain a two-dimensional mapping that consists of two coupled recursion equations. This system determines the motion of the projection of the spin configuration upon the subspace of patterns $\{\xi^\mu\}$ and $\{\xi^\kappa\}$. The method can be trivially generalized to spin configurations strongly overlapping with more than two patterns. As an example, for three macroscopic overlaps $m_1(t)$, $m_2(t)$ and $m_3(t)$, *e.g* the equation for m_2 reads:

$$\begin{aligned} m_2(t+1) = {} & 0.25F\big(\lambda_1 m_1(t) + \lambda_2 m_2(t) + \lambda_3 m_3(t)\big) \\ & + 0.25F\big(-\lambda_1 m_1(t) + \lambda_2 m_2(t) + \lambda_3 m_3(t)\big) \\ & + 0.25F\big(\lambda_1 m_1(t) + \lambda_2 m_2(t) - \lambda_3 m_3(t)\big) \\ & + 0.25F\big(-\lambda_1 m_1(t) + \lambda_2 m_2(t) - \lambda_3 m_3(t)\big). \end{aligned} \tag{7.13}$$

It is not difficult to include external fields modelling a shift of the firing threshold or input from a uniformly firing separate part of the brain; this will be used for a discussion of sleep in Chapter 11.

* If the two patterns are not orthogonal the two terms have to be added with different weigths (Derrida *et al.* 1987).

7.2 General asymmetric models

No dilution, no simplicity. That is one's first impression when one starts to calculate the evolution of overlaps for fully connected Hopfield-type models. It is a pity because results from computer modeling are so easy to obtain, and one would like to understand them.

Theory is hard however. To arrive at just $t = 2$ time steps is a real *tour de force* (Gardner, Derrida and Mottishaw 1987). The results confirm what was already mentioned in Section 4.2: starting with a small but macroscopic overlap with one of the patterns, the overlap grows for at least two time steps, thereby assuring a transient dynamical retrieval of the pattern, even if $\alpha = p/N$ is much larger than the Hopfield overloading value (up to $\alpha \approx 0.64$).

Later on evidence began to accumulate that the Hopfield model has a number of embarrasing properties that make a dynamical analysis difficult, and one can get rid of some of them to achieve progress.

One thing is the symmetry of the coupling coefficients: the effects of a signal sent out in many directions come back in the next time step, and that gives contributions to inputs from many directions which are pronouncedly coherent, far from being statistically independent like in the diluted model.

These coherent terms signalize non-vanishing of the *symmetry parameter*

$$\eta = \frac{\overline{J_{ij} J_{ji}}}{\overline{J_{ij}^2}}, \tag{7.14}$$

where the overline denotes averaging over a distribution of learned patterns (see below). Then one can set out to collect such terms systematically. The *fully asymmetric* case $\eta = 0$ will be a convenient starting point, free of many of the above-mentioned difficulties.

An important guiding principle through the forest of these complications is *gauge invariance* (Toulouse 1977): the property of Ising-like spin models without external fields to have all their static and dynamical properties unchanged if one turns a spin and simultaneously all coupling coefficients connected to this given spin into their negatives. The origin of the name is the analogy with transforming

locally the phase of a Schrödinger wave function and simultaneously the coupled electromagnetic vector potential. The invariance entails that only gauge-invariant combinations of coupling constants can play any role in the dynamical evolution. Since a gauge transform inverts J_{ij} and J_{ji} simultaneously, their product remains unchanged and η is a gauge-invariant quantity. The same holds for the stabilities Δ (equation (4.7)).

There is a growing sub-culture utilizing continuous-time dynamics in conjunction with field-theoretical techniques to exploit these items (Crisanti and Sompolinsky 1988, Rieger *et al.* 1988, Kree and Zippelius 1987). The presentation of this direction is beyond the scope of the present book. We remain on the pedestrian (although perhaps more tiresome) ground of tracing time evolution step by step.

Another difficulty with the Hopfield model is that for simple Hebb-rule learning the stabilities Δ_i^μ of a single pattern (see equation (4.7)) vary from one spin (neuron) to the other (indeed: they have a Gaussian distribution), therefore neurons are not equivalent and averaging over them (as required in (7.1) or (7.2)) is not related simply to averaging over random patterns (no self-averaging can be done: of course this is common to many disordered systems, see Mézard *et al.* 1988). In this respect strict-stability networks in which (6.1) is satisfied as an equality with a Δ independent of the neuron i, like after Adaline learning (Section 6.2), are much handier (see Appendix B).

It is for such strict-stability networks that numerical analysis of the dynamics close to attractors has been recently pushed ahead (Forrest 1988, Kepler and Abbott 1988). The results show that learning algorithms should be tested not only with respect to the storage capacity achieved, but also with respect to *content-addressability*, as measured through the *minimum initial overlap* with a pattern required to retrieve it. Perceptron-like iterative learning algorithms (section 6.2) are superior to the simple Hebb rule not only in providing larger storage capacity, but also in digging much wider basins of attraction. For strict-stability learning rules these basins have 'smooth vertical walls' *i.e.* their boundaries are rather sharply defined if the size of the network is sufficiently large; for small model sizes the boundaries are rounded which also tends to reduce the content-addressability.

For a growing size N the minimum initial overlap is found to differ from unity by $\mathcal{O}(1)$, which means that patterns can be retrieved from an input with $\mathcal{O}(N)$ errors. In other words, the basins of attraction are as big as $\mathcal{O}(N)$, much larger than one might estimate from the $\mathcal{O}(\sqrt{N})$ lower bound (6.4).

Theoretical analysis is still difficult however. To further simplify, when investigating the evolution of the system in the attractor of a single pattern, one can try to replace the effect of all other patterns by just noise, retaining only a few well-controlled statistical properties (Krauth, Mézard and Nadal 1988). This so-called *one-pattern model* can be regarded as a sophisticated elaboration of the signal-to-noise philosophy (Section 4.2). Technically this model eliminates one more complication inherent in the original Hopfield model: since there is only one pattern and noise here, it is easy to check that gauge-invariant combinations of the couplings beyond the stability Δ and the asymmetry η (*e.g.* the triple coupling $\sum_{j,k} J_{ij}J_{jk}J_{ki}$ that would make the calculation much more complicated although not impossible in the many-pattern case) now vanish in the thermodynamical limit $N \to \infty$ at least as fast as $N^{-1/2}$. Due to this fact, the dynamical evolution could be calculated up to 4 time steps; the results are in good agreement with simulations on the original non-randomized network, reproduce the $\mathcal{O}(N)$ size of the basin of attraction, and support the essential correctness of the one-pattern approach.

Some details based on our own work (Pázmándi and Geszti 1989) are presented in Appendix B. In this study the dynamics of a *two-pattern* network, sharing all the simplifying features of the one-pattern case, has been studied by a trick that makes calculations still simpler: in collecting signals echoing on partly symmetric connections, we utilize the fact that these are *individually* small, only add up coherently to a non-negligible contribution. They pass however the non-linear filtering of the sending neurons still individually, which allows one to linearize there. This is the same linearization what we did in Section 4.3 in deriving the static mean-field equations (see also Peretto 1988a).

Our motivation to investigate a two-pattern network with unequal stabilities for the two patterns and the effect of all other patterns randomized like in Krauth *et al.* (1988) was to face the warning from the learning-within-bound model men-

tioned at the end of Chapter 5: in trying to associatively retrieve one pattern, the noise description of all other patterns can fail if one of them has a larger amplitude or stability than the one we are looking at: the stronger attractor can destabilize the weaker one, even if the noise level is not too high.

The application of the formalism outlined above to two-pattern dynamics with patterns of unequal stabilities (acquisition strengths) shows a wide variety of behaviours, depending on the two stability values and on whether we have strict-stability learning or a Hebb-rule-instructed asymmetrically diluted DGZ model. For strict stabilities one observes a re-shaping of the dynamical flow between the two retrieval states and a narrowing of the basin of attraction of the weaker pattern in the presence of the stronger one. For the DGZ model in which the stability is Gaussian-distributed, the effect is more drastic: the weaker pattern may become unstable; in that case the flow from it will be attracted to the stronger one, as anticipated above.

The little collection of solvable-dynamics neural network models has recently enriched by a nice new piece: for a network with many real patterns of strictly constant stabilities Opper *et al.* (1989) noticed that on approaching perfect retrieval of a single pattern, the effective strengths of symmetry-induced coherent terms vanish exponentially. This makes dynamics in the immediate neighbourhood of the pattern as transparent as that of the fully asymmetric models. The reason (or at least one of the reasons) of the simplicity discovered here is that perfect retrieval suppresses the fluctuations allowed by a given non-unit overlap with the pattern.

8 FEED-FORWARD NETWORKS

8.1 The simple perceptron

Each neuron is a classifier. If it receives signals from an "input layer" of neurons (you may think about your retina) whose possible firing patterns $\{\xi\} \equiv \xi_1, \xi_2 \ldots \xi_N$ may carry some meaning, its output

$$S = \text{sign}(\sum_j J_j \xi_j - U) \tag{8.1}$$

(for simplicity we use here hard threshold processing) classifies the input patterns: it is equal to $+1$ for some of them, -1 for others. Which pattern evokes one response and which the other, it depends on the weights J_j ($j = 1 \ldots N$) and the threshold U. It should be noticed that the threshold is equivalent to an extra input from an external neuron fixed to permanent firing $\xi_0 = 1$ and coupled to the operative part of the network through the inhibitory synapse $J_0 = -U$. This can even have some biological relevance (Geszti and Pázmándi 1988; see below in Chapter 11).

The simplest "neural computer", now called a *perceptron* (Rosenblatt 1961), is a single neuron of programmable threshold, connected to an input layer by connection weights that can be programmed too (figure 8.1), so as to solve a given classification task, specified by the desired output for each input pattern. For a set of p patterns $\{\xi^\mu\} = \xi_j^\mu$ ($j = 1 \ldots N$, $\mu = 1 \ldots p$) and a hard threshold function this requires to specify N connection strengths J_j and a threshold U so as to simultaneously satisfy the p conditions

$$\text{sign}(\sum_j J_j \xi_j^\mu - U) = \zeta^\mu, \quad \mu = 1 \ldots p \tag{8.2}$$

where $\zeta^\mu = \pm 1$ is the desired output for the μ-th input pattern.

Instead of classifying by a single binary output variable, to each input pattern one can associate complex output patterns of several binary signals displayed on

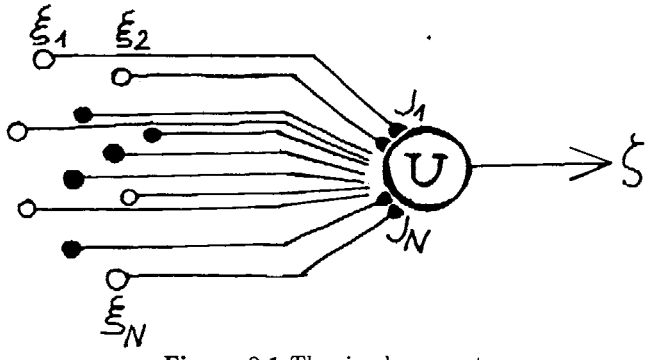


Figure 8.1 The simple perceptron

a screen of output neurons. These output units are however just independent perceptrons fed by the same input layer, and in the following we return to restricting our attention to just one of them.

A simple geometrical interpretation, already present in Minsky and Papert's book (1969), gives enormous help in understanding the classification problem and its more complicated counterparts to be treated later. The framework is this: one should regard a pattern ξ_j^μ with $j = 1 \ldots N$ as a vector of an N-dimensional space. Since each component of this vector is ± 1, each pattern locates a corner of an N-dimensional cube (or "hypercube").

In this space the association task can be depicted in the following way: paint black those corners of the hypercube for which an output $+1$ is desired; white those for -1; corners belonging to none of the input patterns (the overwhelming majority of the 2^N corners) should remain unpainted. Then fix the N connection strengths J_j $(j = 1 \ldots N)$ and the threshold U in such a way that

$$\sum_j J_j \xi_j^\mu \begin{cases} > U & \text{for a black-corner pattern,} \\ < U & \text{for a white-corner one.} \end{cases} \tag{8.3}$$

Now this condition can be satisfied if black and white corners are on the opposite sides of an $N-1$-dimensional plane (of course: a "hyperplane") $\sum_j J_j \xi_j^\mu = U$. To program a perceptron means to find such a plane if possible (for $N = 2$ and a simple classification task see figure 8.2).

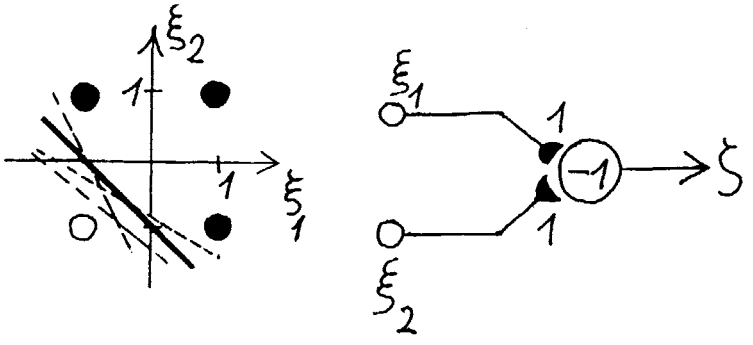


Figure 8.2 The "OR" problem (output +1 if at least one input is +1) and a perceptron that solves it by implementing the solid dividing line. A continuum of other parameter values (dashed lines) solve the problem.

The example in this figure shows that the Gardner (1988) phase-space argument (section 6.4) is not restricted to feedback dynamical networks. Indeed, a given classification task is solved by values of $J_1, \ldots, J_N, U$ restricted to some volume in the $N+1$-dimensional "phase space of connection strengths", the size and shape of which is determined by the nature of the task.* Easy tasks are solved within a big volume, difficult ones within a small one.

And finally: there are the impossible tasks, sometimes of an annoying simplicity. The most famous (or, rather, infamous) example of them is to evaluate the XOR (exclusive-OR) function: a single binary output ζ depending on a two-component binary input ξ_1, ξ_2, specified for four "patterns": (+1,−1) and (−1,+1) should give +1, whereas (+1,+1) and (−1,−1) should give −1. You can check immediately that these requirements are contradictory.

Within the above geometrical picture (figure 8.3) the "deeper reason" is obvious: whatever the weights J_1 and J_2 and the threshold U, no single straight line can be drawn with (+1,−1) and (−1,1) on one side while (+1,+1) and (−1,−1) on the other.

---

* As mentioned above, U can be regarded as one of the connection strengths; namely, that connecting to a fixed input $\xi_0^\mu = -1$.

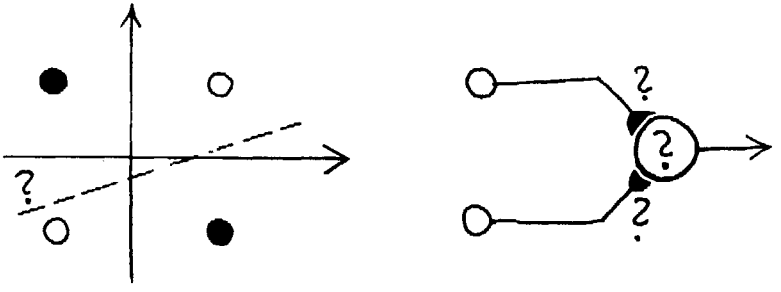


Figure 8.3 The XOR problem: no straight line can divide the black points from the white ones

The general statement is this: a single perceptron unit can do only *linearly separable* classification tasks, i.e. those for which a single plane can be pushed between the black and white corners of the hypercube. Obviously such tasks are more the exception than the rule.

To overcome this difficulty, more complicated networks have to be built from the same blocks. The basic idea is to do processing in several steps, by turning the output of one step into input for the next. Here a variety of different architectures is possible. Two opposite limits reached a degree of maturity in recent years: the layered feed-forward nets in which steps of processing are assigned to subsequent layers of the network, and the fully connected dynamical nets called Hopfield networks where the output-to-input conversion means feedback and the steps are time units in which the same network changes its configuration. The latter was the main subject of this book, however we give a short review of the former in the next section.

So far we concentrated on the solvability of the tasks. However, for the solvable ones, we need algorithms to find a solution.

Here our acquaintances from section 6.2 help: the Perceptron rule for strictly binary inputs and outputs with hard-threshold processing; the Adaline rule for a continuous output. Actually, both of them were originally developed for the simple perceptron. The only departure from the case of autoassociation tasks treated in

section 6.2 is that now the Hebbian correction changes to be added to a connection strength J_j are determined by the input pattern and the desired output. Again, for the Perceptron rule the Perceptron Convergence Theorem (section 6.3) holds, whereas the Adaline rule can be formulated as gradient descent towards a minimum of the cost function

$$W = \frac{1}{2}\sum_{\mu}(\zeta^{\mu} - S^{\mu})^2, \tag{8.4}$$

representing an overall measure of deviations of the actual outputs S^{μ} from the desired ones ζ^{μ}.

What happens if you try to run the perceptron algorithm for an unsolvable classification task? You end up with some errors. If you insist, trying randomly chosen patterns, sometimes you may improve the score. Keeping "in your pocket" a record of the connection strengths corresponding to the smallest number of errors so far obtained (the Pocket Algorithm: Gallant 1987) you obtain a very useful starting point for programming multilayer networks (Mézard and Nadal 1989, Nadal 1989: see the following section). The result can be regarded as finding a plane that bisects the input hypercube so as to have as many black points on one side and white ones on the other as possible. It is worth mentioning that this is a not-least-square optimization problem for which efficient algorithms are not so abundant, thus the neural computing approach may offer new chances.

8.2 Layered feed-forward networks

The XOR function can be calculated by a "hidden layer" of two perceptrons acting in parallel, followed by a single output unit (figure 8.4). This architecture is called a *feed-forward two-layer perceptron*. To guess this possibility, all one has to recognize is that the XOR function is the combination of two subsequent simple logical operations: $XOR = (OR).AND.(not(AND))$. Based on this insight, one can easily construct the neural network carrying out the desired operation. Obviously, an OR gate can be implemented by a low-threshold neuron, an AND-gate by a high-threshold one.

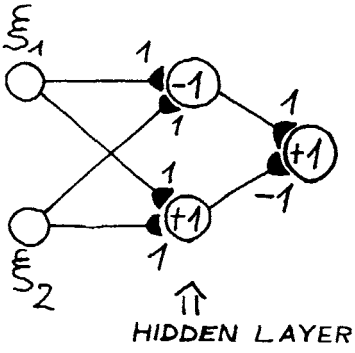


Figure 8.4 A two-layer perceptron for XOR: an OR gate is implemented by $-2 < U < 0$, an AND gate by $0 < U < 2$, a negation by $J_i = -1$.

Minsky and Papert in their famous book on perceptrons (1969) argued that the *insight* into the strategy of solving the problem is absolutely necessary in finding the weights and thresholds, and no simple extension of the Perceptron or Adaline learning rules is available for the multilayer case that would do the job if no such previous understanding is there. Thus a perceptron-like neural network cannot be programmed to do anything but the most trivial association tasks that are probably easier to solve without a neural network.

This very discouraging belief turned Perceptron into a Sleeping Beauty until Parker (1985), LeCun (1985) and Rumelhart *et al.* (1986) found the missing practical algorithm called *back-propagation.*

This algorithm is a straightforward generalization of the Adaline rule in its reformulation as a gradient descent problem. Take the same cost function as for the simple perceptron, equation (8.4); this function now depends on all connection strengths in the various layers through the configurations of the intermediate-layer neurons. Minimize it by gradient descent in the connections to the output layer first, then in the connections to the layer behind it, and so on until the input connections are updated. This is called back-propagation. Iterating the

learning procedure for a cyclic repetition of a given set of input patterns and their desired outputs, one arrives at useful convergence usually in a long but reasonable computer time, although no formal convergence proof exists for the reason already mentioned in section 6.2: Adaline-type learning converges to a local minimum of the cost function; no guarantee that to the global one.

For variants of the back-propagation method and the numerous applications worked out so far the Reader is referred to the PDP (Parallel Distributed Processing) book (Rumelhart and McClelland 1986) and to Hertz and Palmer (1989). Here we mention only the famous NETtalk: the network that could be taught to read aloud by presenting it examples of written text and its correct pronunciation during a couple of hours' computing time, resulting in a performance almost as good as that of another computer program developed by a group of distinguished linguists who trained their understanding of the problem during a ten-year period.

This example illustrates some prominent features. One is that a layered feed-forward network is at its best in learning from examples, in a not-too-intelligent way. However, the patterns of connection strengths resulting from perceptron learning may reveal some marked pathways of processing, representing a sort of hidden insight into a good (if not optimal) strategy. It is of course possible to extract this insight by studying the so-called *internal representations* of the learned patterns: the configurations of firing and quiescent neurons in the intermediate layers. These internal representations are sometimes quite impressive, e.g. single neurons may be responsible to pass the signal to the next layer if the input pattern possesses some particular property. Such single neurons internally representing a feature are sometimes called grandmother-neurons, implying that one of them in your brain would be specialized to fire if and only if you see your grandmother.

Conversely, partial insight into an optimal strategy can be sometimes used to provide a beneficially biased initialization of connection weights before back-propagation is started. It can be advantageous to build this insight into a separate next-to-input preprocessing layer (Denker *et al.* 1987).

An absolutely important feature of perceptron learning is the ability of *generalization*, i.e. the ability of guessing the correct general rule from an incomplete set

of training examples. In the example of NETtalk, it means that after the learning period the network is able to correctly (with a high probability) pronounce text that had not been shown before. In more formal terms it means that our neural computer can be programmed to evaluate a given boolean (binary-valued) function of any input pattern by just showing it a restricted part of the table of that function.

The background of this ability has been beautifully clarified by Carnevali and Patarnello (1987): what can be learned from an incomplete set of examples is *simple* rules. Such rules can be implemented by many different networks, or – in the language of Gardner (1988), see Section 6.4 – in a large volume of the "phase space of connection strengths". This number $N(P)$ of different networks for the same rule P (or this "phase space volume" $V(P)$) can be regarded as a measure of simplicity of the rule (conversely: a small number or small phase space volume of good solutions means high complexity of the rule to be implemented). Now the point is that learning an incomplete set of examples ends up with high probability in just one of the many networks implementing the simplest rule compatible with the examples. This is how generalization is achieved.

Carnavali and Patarnello offer some interesting further interpretation. If $\log(N(P))$ or $\log(V(P))$ is regarded as the entropy of rule P (high entropy means high simplicity or low complexity of the rule) then the above statement can be regarded as a manifestation of the second law of thermodynamics: incomplete learning ends up in the highest-entropy rule compatible with the examples. The same thing expressed in an information-theoretical language says that starting from the maximum-entropy state of complete ignorance, to learn the simplest rule which is the highest-entropy one requires the least amount of entropy reduction i.e. the least information input, thus it can be guessed from the least number of examples.

The back-propagation algorithm brought neural computing into the realm of reality. It is now implemented in various more or less commercial "neural computer" co-processor cards, partly using specialized hardware consisting of one-neuron elementary circuits, partly just simulating the neural network on a serial

processor, applying only a special circuit for the weight multiplications and additions. The first solution has a difficulty with making accurate tunable resistors for the weights; the second does not yet exploit the potential of parallel processing characteristic of biological neural networks. Years to come may show a spectacular development of neural computers. A useful state-of-art survey has been given by Hecht-Nielsen (1988).

However efficient in general, the back-propagation algorithm is certainely a brute-force method that cannot be expected to find any subtle way to solve a particular problem. Developing alternatives is a very active area of research.

A most straightforward objective of improvements is the presence of big flat plateaus on the surface of the cost function W, which makes the driving force $-\partial W/\partial J_{ij}$ very weak and learning very slow. There are suggestions to use different cost functions providing larger gradients. Physicists' contribution is an "entropic cost function" (Solla *et al.* 1988).

An alternative of different character, still of the random search type but introducing a shift of attention towards internal representations, is to take the bits of these representations – the configurations in the hidden layers between a given input-output pair – as the basic dynamical variables, and looking for an optimum in random flips of these variables (Grossman *et al.* 1988, Hertz 1988).

Analogous tasks can be solved by a class of feedback dynamical networks called Boltzmann machines: this will be described below in Chapter 9.

A common drawback of the random search algorithms is that they give no hint whatsoever about how many and how big layers ought to be formed to solve a given task. Failures to converge, usually described as "getting stuck in a local minimum of the cost function", can be also due to a subcritical size of the architecture (Czakó 1989); conversely: oversizing may cause an excessive amount of superfluous learning time.

A few of the recent alternatives (Mézard and Nadal 1989, Ruján and Marchand 1989) made an important progress by amplifying the geometrical view discussed in section 8.1. As a common practical point, they set out to build a layered feed-forward network in which the number of layers and that of neurons in each

layer are determined by the task to be solved, not by vague intuition which was the case for back-propagation.

Let us restrict ourselves to classification tasks, the solution of which is a single binary output (yes or no) to each input pattern. We have to deal mostly with layered feed-forward networks in which each layer is connected only to the preceding and the subsequent one.

The framework common to both of these pioneering approaches is this: let us regard the hypercube of patterns which are input to a given hidden layer. Each neuron in the hidden layer introduces a plane dissecting that hypercube. These planes cut the hypercube into compartments of patterns, called classes. All patterns in the same class have the same representation formed on the hidden layer, therefore the number of different output patterns from that layer is usually less than that of the inputs. This layer-to-layer compression of data is expected to converge to the desired classification in the last layer consisting of a single output neuron.

An obvious requirement is to keep the representations formed in the hidden layers *faithful*: inputs desiring different outputs must be represented by different patterns in each intermediate layer, since otherwise in the strictly layered feed-forward architecture there is no chance to distinguish them later. In the geometrical language this means that each layer must contain a sufficient number of units (neurons) introducing enough dissecting planes until no two patterns desiring different outputs remain in the same class.

From this point the two proposals differ from each other. Mézard and Nadal start building a new layer by checking if the task can be fully solved by a single neuron, now called the "master unit" of the new layer. This is done by the "pocket algorithm" version of the Perceptron rule (section 8.1), speeded up by a judiciously chosen initial state. If it gives the desired classification then the problem is already solved. If not, then the representation offered by this master unit is not even faithful. Therefore the layer must be supplemented by more units one by one (which gives the procedure the name "tiling algorithm"), to section each of its unfaithful classes into faithful parts. This is the hard part of the computation,

however it can be done efficiently and the whole procedure can be proved to converge to no error in a finite number of steps. The proof refers back to the act of constructing the master unit: even before starting the pocket algorithm, an initial set of connection strengths can be chosen so as to reduce the dimensionality of the faithful representation at least by one in each layer.

The approach by Ruján and Marchand envisages to solve the problem by a single hidden layer in which therefore not only a faithful internal representation has to be formed but also a linearly separable one. As these authors realized, this happens if the planes, introduced by the hidden neurons and cutting the hypercube of input patterns into faithful (unicolour) classes, do not intersect each other within the hypercube.

This requirement is not easy to satisfy, however it can be done in a practical way, called the "greedy algorithm." One has to discretize connection strengths and thresholds to obtain a restricted choice of planes, which are then all checked to find the one chopping off the biggest corner of the hypercube. This plane being selected, all other planes intersecting it within the hypercube are omitted from the choice, so are the chopped-off patterns, and the procedure is repeated until full sectioning.

Learning and generalization properties of the new generation of learning algorithms described above are very promising. Apparently, we are only at the beginning of an extensive development of more and more efficient procedures to teach feed-forward neural networks.

9 THE BOLTZMANN MACHINE

The dynamical aspects of the Hopfield model are very attractive; our brain certainly does have dynamics, and a Hamiltonian model with simple attractors has good chances to reveal important mechanisms, as it did for the simple task of associative memory.

In the Hopfield model the question of input and output is treated in a simple way: input is an initial configuration of the whole network; after relaxing to an attractor, output is read off from the configuration of the whole network.

Our nervous system allows for more specialization in this respect. Our sensory organs are input devices (although your eyes may output some important information about your state of mind). On the other hand, our organ of speech as well as our muscles and glands are output devices, however the rest is kept invisible to do the processing. Obviously this has enormous advantages.

The idea of the Boltzmann machine (Ackley *et al.* 1985) is to allow a similar specialization in a Hopfield-like model with symmetric connection strengths and finite-temperature spin dynamics, *i.e.* to distinguish between a group of input neurons, another (possibly partly overlapping with the first) group of output neurons, and the rest of "invisible" processing neurons.

The typical association task for such a network is to create metastable states of the whole network in each of which a given input and its desired output appear together, with any configuration of the invisible units. Once such associative metastable states learned by appropriate modifications of the synaptic strengths, the association is carried out by just fixing the input, which suffices to force the network to relax into the corresponding metastable state. Finally, the desired output appears in the place it was expected.

How to do the synaptic modifications? Hebb's rule gives a hint: for a given connection strength J_{ij} one has to observe the products of spin variables on both ends of that connection; however, since we are working in a fluctuating environment, a "thermal" average $\langle S_i S_j \rangle$ should be taken.

How would such a quantity signalize whether the association is achieved? The trick is to clamp the input spins to a given input, and calculate the average twice: firstly clamping output spins to the desired output, secondly letting output spins freely flip. If the two cases give the same result, the association is already done; if not, you correct the coupling strength: denoting one of the connected I-O pairs by I_α and O_α, do

$$J_{ij} \to J_{ij} + \lambda\left[\langle S_i S_j\rangle_{I_\alpha,O_\alpha} - \langle S_i S_j\rangle_{I_\alpha}\right],$$

λ being a learning amplitude. In practice one calculates the desired averages $\langle S_i S_j\rangle_{I_\alpha,O_\alpha}$ and $\langle S_i S_j\rangle_{I_\alpha}$ for all connection strengths and all I-O pairs (all α's), then sums up all the above corrections in one step. This step should be repeated until convergence*.

The calculation of the averages to a reasonable accuracy takes long Monte Carlo runs, and apparently many of them. This gives the Boltzmann machine the bad reputation of being terribly slow. Anyway, it works, and its performances are comparable to – or in some cases even better than – those of backpropagation-taught feed-forward networks.

However for the Boltzmann machine one can do something better: instead of the long Monte Carlo runs, approximate $\langle S_i S_j\rangle$ by $\langle S_i\rangle\langle S_j\rangle$ and calculate $\langle S_i\rangle$ from the mean-field equations (4.19) for a given set of connection strengths (Peterson and Anderson 1987); then do the Boltzmann machine corrections and repeat the procedure until convergence like above. Although the numerical solution of the big system of coupled mean-field equations is not very fast either, the authors claim that it is about 30 times faster than the Monte Carlo procedure. Even if this big gain factor is true only for selected tasks, and the possibility of getting stuck in the wrong solution is still there, the mean-field version certainely makes the Boltzmann machine a lot more competitive.

---

* Instead of the hand-waving argument given above, the same rule can be derived also by gradient-descent minimizing an entropic cost function (see Hertz and Palmer 1989).

For many input units and few output ones, feedback from the latter hardly influences the dynamics of the system and the Boltzmann machine works in an essentially feed-forward mode of operation (Hopfield 1987), which suggests affinity with the Perceptron architecture and the possibility of many workable intermediate forms between the two.

However much has been gained in applications with respect to the original Hopfield model, the theoretical simplicity is apparently gone. A search for restricted but theoretically solvable Boltzmann machines would be welcome.

10 SELF-ORGANIZED FEATURE MAPS: THE KOHONEN MODEL

10.1 Preliminaries

The original paradigm of perceptron programming is *supervised learning*: we specify a desired output to be associated to a given input, at each stage of the learning process check the distance from the aim, and conduct changes so as to get closer to it.

Supervised learning is not restricted to the presence of a personalizable supervisor like a parent or a teacher. To seize an object by hand is also learned by an error correction mechanism that converges to associate an accurately defined motor output to a given visual input.

This example however opens a window to other aspects of the problem, reaching beyond supervised learning. The new aspects are connected to the relative independence of the intermediate levels of processing.

In the above example one should distinguish at least two such intermediate levels. One is visible: the degrees of freedom of your joints allow a broad choice of the ways you can reach out for an object. The other is somewhat more abstract: for the sensory input an internal representation has to be formed in your brain, then translated into a motor output. Although the final aim of the association is well determined, there is large freedom in choosing the internal representations.

In case of too much freedom, your brain – or, if you want to build an electronic implementation, your robot control set – can do different things. One is to choose just one feasible solution by an algorithm that is straightforward enough to go the right way: an example is back-propagation with all its alternatives.

The more demanding but more promising alternative, apparently widely exploited by our brain, is to optimize according to some global organizing principle: this can be called self-organization. For the externally visible part of the above example, the principle can be some global minimization of muscle and joint stresses,

the result of which can be given an impressive mathematical formulation (Pellionisz 1988).

For the internal representation stage the global principle seems to be *topological organization*. This means that whenever it makes sense to speak of *neighbouring* inputs (e.g. neighbouring directions in what you see, or neighbouring pitches in what you hear), the internal representations are formed so that neighbouring neurons respond to neighbouring inputs. A necessary prerequisite is of course to provide a *localized* response to sharply localized inputs.

For the enormous amount of information to be encoded and restored in visual processing, topological organization seems to be the only chance. This chance has been embraced, which is an outstanding achievement of biological evolution.

Self-organization in early infancy begins as a rough elimination of part of the existing synaptic connections and creating new ones, and continues as a fine tuning of synaptic strengths. To do all this by self-organizing is absolutely necessary for at least two reasons: there is no genetic capacity to store the enormous amount of information needed for the tuning, and Nature cannot foresee the features of the particular environments an individuum has to adapt itself to. This ability of adaptation through self-organization should be preserved for the whole life, e.g. in order to give more resolution to input fields containing more detail, and also to correct lesions and deterioration on aging. Electronic implementations of neural networks envisage the same property.

From the consideration of visual processing one more simple conclusion seems to emerge. It is that the layered feed-forward architecture actually present, its processing layers being divided between the retina and the visual cortex, is a natural way of achieving the fast processing needed, and the after-birth period is apparently used for a first programming of this feed-forward network. The highly localized representations needed for topological self-organization are then naturally processed along well-organized feed-forward neural pathways.

Without being too specific, the formation of a layered architecture during natural evolution is not more difficult to imagine than that of anything in our body; now it is part of our genetic heritage. Less trivially, so is a simple token of "neural

computing" that assures locality without further learning: this is the mechanism of *lateral inhibition* through an ancillary infra-structure of extra inhibitory connections *within* a layer (figure 10.1): the neurons of a layer receiving a localized input would compete among themselves for a few time steps, until most of them are suppressed and only a "winner" would transmit the signal.

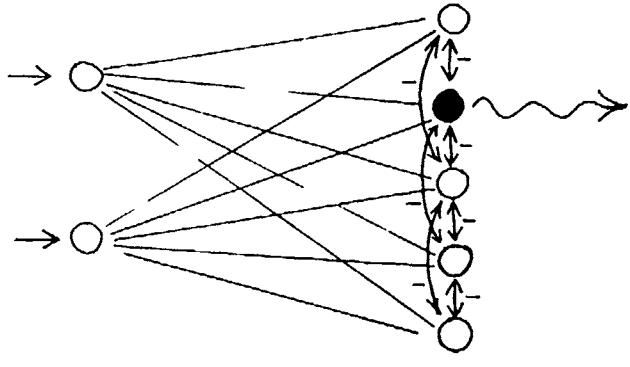


Figure 10.1 **Feed-forward network with in-layer lateral inhibition: only the winner fires**

It is important to notice that on this point we entered the study of networks of higher complexity than considered so far: from here on we have to do with a feed-forward network composed of feedback dynamical layers.

10.2 Topological ordering: local and global

Local representations are necessary but not sufficient to successfully handle the wealth of visual information. What is still missing is *topological ordering*: inputs from neighbouring directions of the visual field should excite neighbouring neurons in the processing layers.

If this local topological ordering continues over the whole layer without any "topological defect" (figure 10.2) then the layer is globally ordered: scanning the

visual field along any straight line would excite a non-self-intersecting line of neurons in the layer. Such an organization is apparently quite advantageous for information processing.

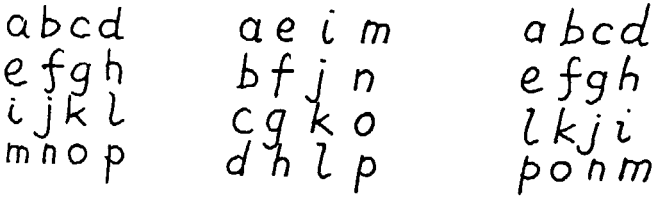


Figure 10.2 a) A screen of input points; b) one of its completely ordered mappings (other possible solutions are related by reflexions); c) a mapping with a topological defect (a twist)

It was recognized by Grossberg (1976) and by Willshaw and von der Malsburg (1976) that a slight refinement of the lateral inhibition mechanism, which is actually present in our brain, suffices to achieve *local* topological ordering. This is the so-called "Mexican hat" arrangement of in-layer connections (figure 10.3): the closest neighbours of a firing neuron are excited, the more remote ones are inhibited.*

The important point about the Mexican hat is that it gives neighbouring neurons the tendency to fire together. Then, just by Hebbian learning, the active feed-forward inter-layer connections incoming to neighbouring neurons become gradually similar to each other, and this teaches them to respond to neighbouring inputs. This is how *local* topological ordering comes about.

The great surprise follows here: local ordering, which arose in a transparent way from intra-layer Mexican hat + inter-layer Hebbian learning, entails also global ordering by a slower process. The details of this process are our concern below.

To sum up where we are now: just a few simple and evident-looking principles, genetically inherited (layered feed-forward architecture adaptable by Hebbian

* Convenient mathematical representations of the Mexican hat are the difference of two Gaussians, or – less ambitiously – of two square wells.

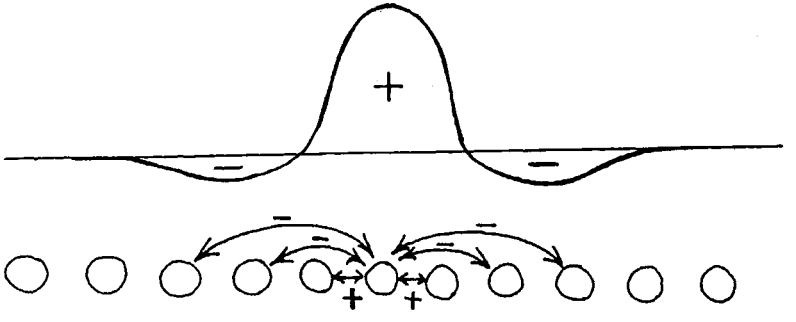


Figure 10.3 In-layer connection strengths are excitatory for short distances and inhibitory for longer ones: this is the "Mexican hat"

learning, supported by an infra-structure of in-layer connections of the lateral inhibition or Mexican hat type and), suffice to assure self-organization to a *topologically correct mapping* of input fields onto layers of processing neurons.

Such mappings onto various areas of the quasi-two-dimensional cortex are actually the way our brain processes various kinds of information. For some details see Knudsen *et al.* (1987) and Kohonen (1988).

10.3 The Kohonen algorithm

Kohonen (1982) introduced a simple and computationally very efficient algorithm that compresses all the complexities of Mexican hats and Hebbian learning into a simple formula implementing local topological ordering.

The Kohonen network is a neural computer that selects the "winner", the only one firing out of a layer of neurons, each receiving the same vector $\vec{x} = \{x_j\}$ (a "feature") of inputs multiplexed for them by $j = 1 \ldots M$ input neurons. The input feature is mapped onto the output of the winner: this is the *Kohonen feature map.*

Let the neuron index in the layer be a vector $\mathbf{i}$, with $|\mathbf{i} - \mathbf{i}'|$ measuring the physical distance between two neurons. Selecting the winner, according to equation

(2.2) with a continuous output function $f(X)$, means to find neuron $\mathbf{i} = \mathbf{i}^*$ for which $\sum_j J_{\mathbf{i}j} x_j \equiv \vec{J}_{\mathbf{i}} \cdot \vec{x}$ is the largest*.

If the connection strength vector is normalized by a spherical constraint like (6.3) fixing $|\vec{J}_{\mathbf{i}}|$, the largest scalar product is given by the neuron whose connection strength vector $\vec{J}_{\mathbf{i}}$ is the most parallel to the input $\vec{x}$. To simplify further, let us choose units in which $|\vec{J}_{\mathbf{i}}| = |\vec{x}|$: then "most parallel" is "closest", and the winner $\mathbf{i}^*$ to which input $\vec{x}$ is mapped will be selected by the condition

$$\mathbf{i}^*: \quad |\vec{J}_{\mathbf{i}^*} - \vec{x}| \text{ is minimum of } |\vec{J}_{\mathbf{i}} - \vec{x}|. \tag{10.1}$$

The Kohonen algorithm is a procedure to adapt the connection strengths $J_{\mathbf{i}j}$ so as to make this mapping $\vec{x} \Rightarrow \mathbf{i}^*$ topologically ordered. This is done in two steps:

- take a random input $\vec{x}$ and determine the corresponding $\mathbf{i}^*$ by (10.1);
- modify connection strengths by

$$J_{\mathbf{i}j} \to J_{\mathbf{i}j} + \lambda(\mathbf{i} - \mathbf{i}^*)(x_j - J_{\mathbf{i}j}), \tag{10.2}$$

where the function $\lambda(\vec{r})$ usually depends only on $|\vec{r}|$, being sharply peaked for small values of the distance r, giving largest amplitude of adaptation to $\vec{i}^*$, then next largest to its closest neighbours, etc. This would bring $\mathbf{i}^*$ and its neighbours closer to $\vec{x}$, which is an efficient way to bring them close to one another.

Steps (10.1) and (10.2) are repeated with random inputs $\vec{x}$ chosen from a given distribution representing the relative importance of different features of the input set, until full ordering and smoothing of the input-to-output map is achieved. For the latter, as iteration goes on, $\lambda(\vec{r})$ should be gradually decreased and its range shrunken to zero. This latter stage is also very interesting; analysis shows that the decrease of λ in processing time should be very slow to assure flawless smoothing (Ritter and Schulten 1988). Here however our main interest is topological ordering, therefore we will regard $\lambda(\vec{r})$ constant in time.

* We use the vector notation of section 6.3, here however $\mathbf{i}$ is an important index since the story is about coupled $\mathbf{i}$-s.

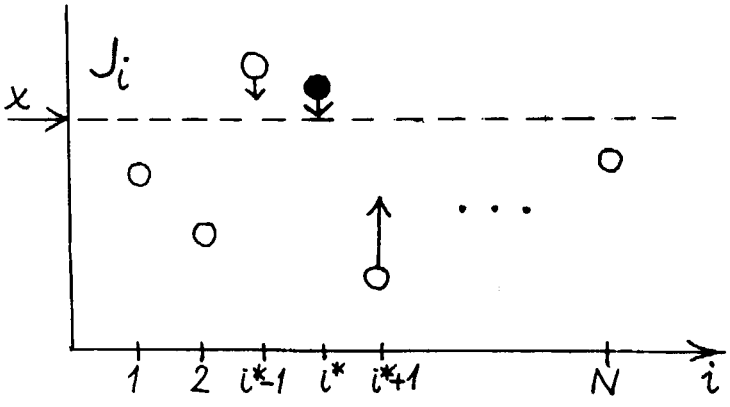


Figure 10.4 The algorithm of the one-dimensional Kohonen map

For a better insight let us have a closer look at the case of a one-dimensional chain of neurons, with connection strengths J_i to a single input x (figure 10.4). Although for this case the route through equation (2.2) and spherical normalization is not feasible, we simply adopt the one-dimensional versions of (10.1) and (10.2):

$$i^*: \quad |J_{i^*} - x| \text{ is minimum of } |J_i - x|; \tag{10.3}$$

$$J_i \to J_i + \lambda(i - i^*)(x - J_i). \tag{10.4}$$

This, if repeated with random inputs x until convergence, is expected to result in a topologically ordered representation of inputs x from some interval, utilizing the whole chain of neurons.

Global ordering in the one-dimensional case would mean to modify the connection strengths so that the index of the responding neuron should become a *monotonous* function of the input x.* The number of neurons responding to a

* Monotonous change can be either growing or decreasing; for a physicist, this is a case of "spontaneous breaking of symmetry".

given interval of inputs is larger where random inputs come more frequently; this is important for the applications but none of our main concerns in this short presentation.

The way the algorithm achieves monotonous ordering can be understood in a formal way as follows (Kohonen 1982; Ritter and Schulten 1986): the Kohonen process is Markovian with negative eigenvalues, having the two oppositely ordered states as the only eigenstates of zero eigenvalue; one of these states is thereby reached and retained indefinitely.

Cottrell and Forth offer a more detailed description of the mechanism of topological ordering (i.e. the establishment of monotonous ordering of J_i in the index i). The new element in the discussion is to observe that a region once ordered, it remains so in a given step of (10.3-4) if the winner and its neighbours to be changed all belong to the same region. This is a trivial consequence of the monotonous decrease of $\lambda(r)$ with r.

Changes can occur however on the boundary between two ordered regions and one of its neighbours (figure 10.5): if the boundary neuron is next-to-winner, it can be pulled over to leave the other-side neighbour as the new boundary.

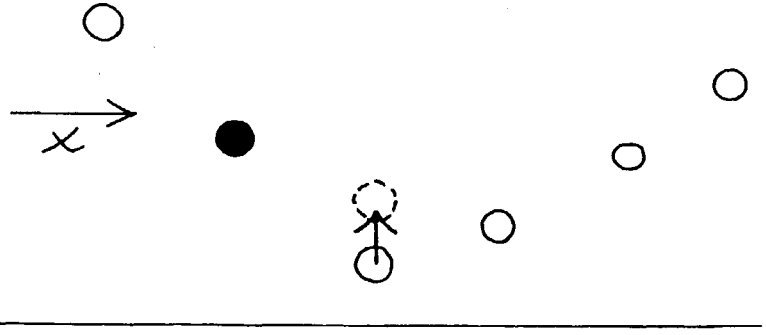


Figure 10.5 The elementary step of boundary motion for the one-dimensional Kohonen map

By this elementary act of ordering, the boundary between the two regions is shifted by one step away from i^*. One can regard global topological ordering in the Kohonen map as a succession of such steps of boundary motion, until all boundaries leave the sample which remains ordered.

Let us try to use this simple boundary motion mechanism to obtain some insight into the ordering dynamics of the Kohonen map (Geszti and Csabai 1989). Our first guess might be that the boundary motion can be treated approximately as a random walk and annihilation process. Adopting this view, since the learning amplitude λ is an even function of the difference $i-i^*$, i.e. it depends only on $|i-i^*|$, boundary steps to both directions are equally probable, therefore the boundaries are moving by diffusion to each other (which is analogous to recrystallization of a polycrystalline material) or towards the edges of the sample until the last of them quits.

On the basis of this reasoning one can try to obtain a simple estimate of the size dependence of the time needed for topological ordering. In a linear sample of N neurons the last boundary can quit the sample in $\mathcal{O}(N^2)$ steps of random walk. On the average $\mathcal{O}(N)$ steps of algorithm (10.3-4) are needed for one step of the boundary, since i^* has to get into the right position to assist the jump. Consequently, the whole topological ordering should take $\mathcal{O}(N^3)$ steps of the Kohonen algorithm*.

The slowness of diffusion, especially for big samples, gives one the idea to make the learning function $\lambda(i-i^*)$ asymmetric, thereby driving the boundaries in a given direction. We call this version the *forced Kohonen map*. In this case the above reasoning suggests that with drifting instead of diffusing boundaries only $\mathcal{O}(N)$ steps are needed for the last boundary to quit the sample, thus full topological ordering should take only $\mathcal{O}(N^2)$ time steps.

Computer simulations roughly support the above expectations: for a symmetric learning function the time needed for ordering is indeed roughly proportional to N^3, which is speeded up with asymmetry to $\mathcal{O}(N^2)$. Nevertheless, the simple reasoning used to obtain the result is not valid, and even for a symmetric learning function

* The actual computing time contains one more factor of N for the time needed to find the winner on a serial computer.

there is a systematic driving force towards ordering. The nature of this force will be discussed in the next section.

More complicated is the two-dimensional case which is far more important for the applications. Here the boundaries between ordered regions are line defects. Ordering itself consists in making two connection strengths $J_1(i,j)$ and $J_2(i,j)$ monotonous functions of both arguments i and j. By spontaneous symmetry breaking, this can be done in 16 different ways. To analyze topological defects of such a complicated ordering must be a heavy exercise in algebraic topology (Mermin 1979).

Simulations show that a single kind of defect dominates the time needed for topological ordering: the one obtained if two opposite edges of the sample order in the opposite way. On the $i(J_1, J_2)$, $j(J_1, J_2)$ plot introduced by Kohonen this defect appears as a twist. For the inverse, this is a saddle point on only one of the surfaces $J_1(i,j)$ and $J_2(i,j)$, the other having been already ordered for most of the time (figure 10.6).

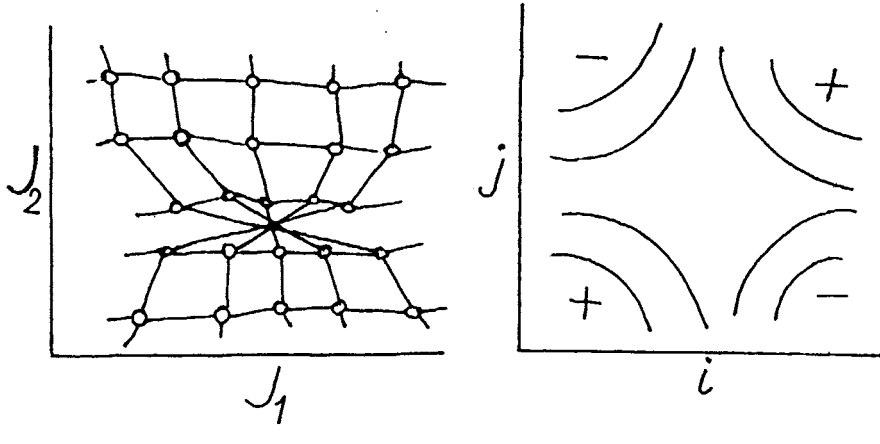


Figure 10.6 The dominant defect in the two-dimensional case, in two equivalent plots. The net on the left has those pairs of connection strength vectors connected which belong to neighbouring neurons. Curves on the right connect neurons of the same value of $J_1(i,j)$.

This time again, introducing asymmetries may help, however for a reason quite different from the one-dimensional case. What should be struggled against here is curvature of lines along which neurons are monotonously ordered. Algorithm (10.2) with a $\lambda(\vec{r})$ independent of the direction is not very sensitive to such curvatures. Now it is uniaxial anisotropy that may improve the situation. Simulations are not yet conclusive on this point; there seems to be a tendency that with such an anisotropy the resultant twist (in a more fancy language: the total topological charge) eventually present in the initial configuration would be swept out more often before getting into a well equilibrated position after which it takes a very long time to get rid of it. Another suggestion now being tested is to make $\lambda(\vec{r})$ depend also on the position of the winner, thereby providing a systematic driving force for topological defects.

10.4 Question marks on theory

As already mentioned above, the diffusion picture of boundary motions is not correct. The reason is that for the last stage of topological ordering with one boundary left (figure 10.7), there is a systematic (although very weak) driving force to sweep the boundary out, even without any built-in asymmetry.

The origin of this driving force is in the asymmetry between the two sides of the almost-ordered configuration. A fluctuational displacement of the boundary can happen in both directions. Kohonen's algorithm however tends to reduce the slopes of J_i as a function of i in each ordered region. This tendency cannot be obeyed in the region stretching between the biggest and the smallest J_i (one of which is the boundary), since those ends are "anchored" to the edges 1 and 0 of the input interval (indeed, if one of the two extremal J_is tries to get away from the respective value 1 or 0, the probability grows for an input to hit the respective edge region and pull it back). Therefore the slope can get smaller only between the boundary and the non-extremal edge of the sample. This promotes boundary displacements according to figure 10.5 towards that edge.

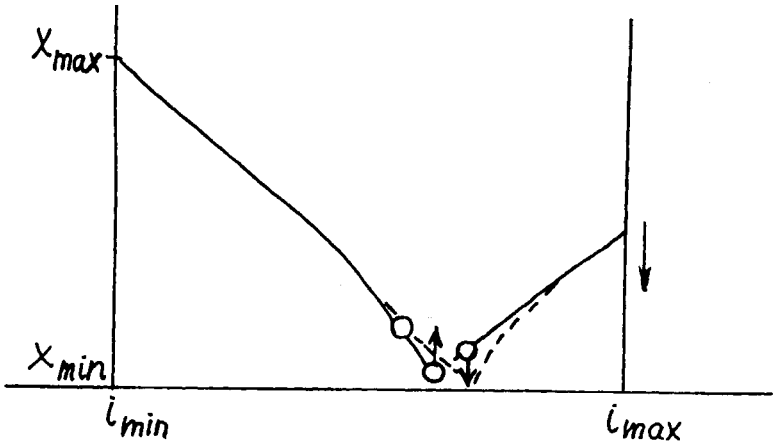


Figure 10.7 The final stage of topological ordering in one dimension

The correct theoretical treatment of this qualitative picture remains to be worked out. What seems to be well established is that a description averaged over the random input x is reasonable for the ordered regions. After some transient time this description can take the form of a nonlinear diffusion equation for the smooth function $J(i) \equiv J_i$, with some correction for mutual "screening" of the two branches competing for the input. On the other hand, the motion of the boundary (the topological defect) is fluctuation-dominated for all times and has to be followed by a detailed "microscopic" consideration of the elementary step (figure 10.5). The hard point is how to describe the coupling of ordered and defect subsystems. The nonlinear diffusion equation referring to the first seems to support the observed N^3 scaling of the ordering time, however for the time being no reliable statement is available about the way this can be modified by the second.

The two-dimensional case raises further difficulties; an important harmful effect seems to be that twisted configurations are stabilized by opposite principal curvatures, eliminating the local driving force of diffusion-like dynamics.

Beside being a useful tool for neural computation (see *e.g.* Kohonen 1988, Ritter *et al.* 1989), the Kohonen map is an exciting dynamical system. Its further study may reveal a lot of good physics.

11 OUTLOOK

Neural network modelling is a lot of fun for those who do it. What else? From here on, opinions diverge.

It is perhaps the question of computing applications on which it is easier to give a fair account. Neural computer systems installed here and there, even in their present-day rudimentary form, solve tasks to the satisfaction of customers. They have their own flavour among the competitors: they attack the problem in what seems a human way, often more transparent for the user than other powerful computational tools. The range of problems they are well suited for is rather limited for the moment, and it is not clear how deep new insight is needed before this may change. Besides, unforseen advances of non-neural computation can deeply influence the issue.

Concerning the great advanture of getting closer to an understanding of how our brain works, scepticism is more widespread. It is however probably fair to say that modelling offers a multitude of metaphores for brain functions, competing for survival on the score that some of them may prove more than a metaphore.

Physicists' models are not among the least fit in the competition. This refers in particular to what we call Hopfield's model because – although many of its features appeared earlier in the literature – its precisely defined setup allowed an unprecedented insight into the details of this particular dynamical system. Based on this insight, modifications and alternatives can now be treated and adapted to a particular application with some ease.

In Chapter 5 we have already discussed the possible connection between short-term memory and the learning-within-bound model. Another early attempt to attach physiological meaning to some feature in the models was a proposed explanation of the possible role of dream sleep by Crick and Mitchison (1983) and Hopfield *et al.* (1983), who argued that unlearning (cf. section 6.2) of overly strong spurious memories retrieved by association to random inputs can be an indispensable act of refreshing the memory. Less specifically, sleep as a switch into a different regime of complexity by a change of the overall level of activity by an "external field" of inputs from another part of the brain, has been illustrated on the solutions of the system of equations (7.11) (Geszti and Pázmándi 1989).

In attacking problems with a biological or psychological motivation, some characteristic pathways begin to take shape. One is to add detail to the models, *e.g.* by working out more carefully the distinction between various kinds of neurons.

Although this is natural, one can foresee from time to time a return to simplicity on a higher level, perhaps by recognizing that some of the details carry a well-defined simple function that – once understood – can be put back as a parameter (like low overall activity, representing the action of a population of inhibitory neurons).

Another characteristic direction is to regard the existing simple models (Hopfield, layered perceptron, Kohonen map) as building blocks and investigate whether their cooperation enables them to solve tasks of somewhat higher complexity. Since our brain in fact consists of a number of anatomically different blocks, this is a natural direction to look. As an example, psychological experiments on scanning the memory to detect the presence of an item, could be interpreted in a natural way as the joint action of three Hopfield-type attractor networks (Amit *et al.* 1989).

This brings us to the realm of looking for an understanding of cognitive functions in terms of neural network models. Here, for the moment, everything becomes a metaphore. To regard cognitive action as a higher computing language based on the machine language of the neuronal processors, probably catches some of the truth. However, as fundamental issues as the machine code itself (for which the frequency or binary code is most probably not the whole story), and the number of levels that may exist above that code, are quite uncertain for the moment.

The feeling of physicists working in this field is that our experience with complex systems can bring us a few steps closer to the nature of these formidable questions.

APPENDIX A

THE REPLICA TRICK WAY TO MEAN FIELD

In this presentation we follow Amit, Gutfreund and Sompolinsky (1977), who adapted this method from Kirkpatrick and Sherrington (1978). The basic idea is the following. We want to calculate thermal equilibrium values of the order parameters; in particular: the overlap with a retrieved pattern. These equilibrium values minimize the free energy of the system that can be calculated by statistical mechanics. This is difficult for a system with frozen-in (quenched) disorder, which in the present case stems from the patterns stored in the coupling strengths by Hebb-rule learning. The reason is that to get rid of the uninteresting complexity of each individual set of patterns, the free energy has to be averaged over the distribution of randomly chosen patterns:

$$F = -\beta^{-1}\langle\langle \ln Z \rangle\rangle_{\xi}, \tag{A.1}$$

where

$$Z = \mathrm{Tr}_S e^{-\beta H(S;\xi)} \tag{A.2}$$

is the partition function of the system, and Tr denotes the trace operation that in the present case means a summation over all possible values of all spins. The (quasi-)energy that in this Appendix we denote by H for Hamiltonian to emphasize that this is now pure physics, depends on the stored patterns; these are fixed (because quenched) when calculating the partition function (A.2) but we have to average over them in the last step (A.1).

It turns out that averaging the logarithm is extremely hard. What is much easier is to average a power of Z, from which however the logarithm can be obtained using the limit

$$\ln x = \lim_{n \to 0} \frac{x^n - 1}{n}, \tag{A.3}$$

which gives the free energy per spin averaged over the quenched patterns in the form

$$f = \frac{F}{N} = \lim_{n\to 0} \lim_{N\to\infty} \frac{-1}{\beta N n} \frac{\langle\langle Z^n \rangle\rangle_\xi - 1}{n}. \qquad (A.4)$$

Now Z^n can be interpreted as the partition function of n copies of the system: the famous *replicas*, which are free in their thermal fluctuations, not being coupled by the energy E, nevertheless they are correlated because they each have the same choice of quenched random patterns, like at two tables of tournament bridge. The correlations between replicas of different thermal histories caused by equal couplings are the main subject of the method; its mathematical strangeness is that these correlations have to be evaluated when the number of replicas goes (by analytical continuation) to zero! A polite embarrassement over this strangeness is expressed by calling all this the replica "trick".

In carrying out the program a long way has to be covered. Let us state the starting point to see why replicas are helpful at all. For the big system consisting of n replicas of a network of N spins (neurons), each replica storing the same $p = \alpha N$ independent random binary patterns, spins belonging to different replicas should be distinguished by a replica index $\rho = 1 \ldots n$. The energy (4.3) with the Hebb-rule connection strengths (3.1) and the convention $\lambda = N^{-1}$ becomes

$$H = -\frac{1}{2N} \sum_{ij\mu\rho} (\xi_i^\mu S_i^\rho)(\xi_j^\mu S_j^\rho) + \frac{1}{2} pn - \sum_{\nu=1}^{s} h_\nu \sum_{i\rho} \xi_i^\nu S_i^\rho, \qquad (A.5)$$

where external fields $h_\nu \xi_i^\nu$ have been introduced to support condensation into a finite, small number s of modes we expect to take an active part in ordering ($s = 1$ for retrieval; $s > 1$ for spurious states).

The fourfold sum can be written in the form $\sum_{\mu\rho} a_{\mu\rho}^2$ whith $a_{\mu\rho} = \sum_i \xi_i^\mu S_i^\rho$. Then – denoting other terms in (A.5) by dots for a moment –

$$e^{-\beta H} = e^{\lambda \sum_{\mu\rho} a_{\mu\rho}^2 - \ldots} \qquad (A.6)$$

where $\lambda = \beta/2N$. This has to be averaged over the distribution of different random patterns, which is hard because of the sum of quadratic functions in the exponent.

Here comes a great trick called the *Gaussian transform* (occasionally also mentioned under the more imposing name "Hubbard-Stratonovich transform"): factorizing the right-hand side in μ and ρ, in each factor just use the identity

$$e^{\lambda a^2} = \int \frac{dx}{\sqrt{2\pi}} e^{-\frac{x^2}{2}+\sqrt{2\lambda}ax} \tag{A.7}$$

with a rescaled variable $x = m_{\mu\rho}\sqrt{N\beta}$ (expressing the knowledge that if we don't do that rescaling, all interesting things in the integral will happen in an interval of $\mathcal{O}(\sqrt{N\beta})$), to give factors of

$$(N\beta)^{\frac{1}{2}} \int \frac{dm_{\mu\rho}}{\sqrt{2\pi}} e^{-\frac{m_{\mu\rho}^2}{2/N\beta}} \cdot e^{\beta m_{\mu\rho} a_{\mu\rho}}. \tag{A.8}$$

This is our first point of rest in the calculation, a result that carries some meaning: we have reduced the problem with a quadratic hamiltonian to one with the linear $a_{\mu\rho}$ in the exponent, at the expense that its amplitude $m_{\mu\rho}$ is now a fluctuating quantity (a "multiplicative noise") of Gaussian distribution, to be averaged over. Later on this averaging will cause troubles to be overcome by saddle-point integration, here however nothing can prevent us doing the desired average over all independent pattern components $\xi_i^\mu = \pm 1$. The result will factorize in the indices μ (independent patterns) and i (decoupled spins) but not in ρ (correlated replicas):

$$\begin{aligned} \left\langle\left\langle e^{\beta \sum_{\mu\rho} m_{\mu\rho} a_{\mu\rho}} \right\rangle\right\rangle_{\xi_i^\mu=\pm 1} &= \left\langle\left\langle e^{\beta \sum_{\mu\rho} m_{\mu\rho} \sum_i \xi_i^\mu S_i^\rho} \right\rangle\right\rangle_{\xi_i^\mu=\pm 1} \\ &= \prod_{\mu i} \cosh(\beta \sum_\rho m_{\mu\rho} S_i^\rho) \\ &= \exp\Big(\sum_{\mu i} \ln\cosh(\beta \sum_\rho m_{\mu\rho} S_i^\rho)\Big). \end{aligned} \tag{A.9}$$

Alas, averaging left us once more with a complicated non-linear function of the spin variables, and we have still the trace to be done. How much simpler it would be if the function in the exponent were just quadratic!

A little magic insight into what one expects to obtain at the end helps here: in an hour's time $m_{\mu\rho}$ will turn out to be the overlap with pattern μ in replica ρ.

Then however it is worthwhile to distinguish the *non-condensed* components $\mu > s$ for which this overlap is $\ll 1$ and expand for them $\ln\cosh x \approx x^2/2$ to obtain for these factors

$$\exp\frac{1}{2}\sum_{\mu i}^{(s<\mu\leq p)}\beta^2\sum_{\rho\sigma}m_{\mu\rho}m_{\mu\sigma}S_i^\rho S_i^\sigma, \tag{A.10}$$

while leave the $\mu = 1\ldots s$ factors corresponding to the eventually condensed patterns ("low patterns") unchanged.

Let us summarize what we have achieved so far (it will not be short):

$$\begin{aligned}\langle\langle Z^n\rangle\rangle &= e^{-\frac{\beta}{2}pn}\,\mathrm{Tr}_{S^\rho}\int\prod_{\mu\rho}\frac{dm_{\mu\rho}}{\sqrt{2\pi}}\\ &\exp\{N\beta[-\frac{1}{2}\sum_{\mu\rho}^{(\mu>s)}m_{\mu\rho}^2+\frac{\beta}{2N}\sum_{\mu i}^{(\mu>s)}\sum_{\rho\sigma}m_{\mu\rho}m_{\mu\sigma}S_i^\rho S_i^\sigma]\}\\ &\times\langle\langle\exp\{N\beta[-\frac{1}{2}\sum_{\nu\rho}^{(\nu\leq s)}m_{\nu\rho}^2+\sum_{\nu\rho}^{(\nu\leq s)}(m_{\nu\rho}+h_\nu)\frac{1}{N}\sum_i\xi_i^\nu S_i^\rho]\}\rangle\rangle_{\{\xi_i^\nu\}}\,.\end{aligned} \tag{A.11}$$

One little rewriting makes the formula more intelligent-looking: in the second sum of the first exponent the $\sigma = \rho$ terms can be included in the first sum; what remains, will contain the overlap $q_{\rho\sigma}$ of *different* replicas ρ and σ:

$$q_{\rho\sigma} = (1-\delta_{\rho\sigma})\frac{1}{N}\sum_i S_i^\rho S_i^\sigma. \tag{A.12}$$

With this definition (A.11) becomes

$$\begin{aligned}\langle\langle Z^n\rangle\rangle &= e^{-\frac{\beta}{2}pn}\,\mathrm{Tr}_{S^\rho}\int\prod_{\mu\rho}\frac{dm_{\mu\rho}}{\sqrt{2\pi}}\\ &\exp\Big\{-\frac{\beta N}{2}\sum_{\mu>s}[(1-\beta)\sum_\rho m_{\mu\rho}^2-\beta\sum_{\rho\sigma}m_{\mu\rho}m_{\mu\sigma}q_{\rho\sigma}]\Big\}\\ &\times\langle\langle\exp\{N\beta[-\frac{1}{2}\sum_{\nu\rho}^{(\nu\leq s)}m_{\nu\rho}^2+\sum_{\nu\rho}^{(\nu\leq s)}(m_{\nu\rho}+h_\nu)\frac{1}{N}\sum_i\xi_i^\nu S_i^\rho]\}\rangle\rangle_{\{\xi_i^\nu\}}\,.\end{aligned} \tag{A.13}$$

Let us concentrate now on the first exponential containing the non-condensed (or "high") patterns $\mu > s$. The quantity $q_{\rho\sigma}$, through its definition (A.12), contains in a quadratic combination the spin variables over which the trace has to be taken. Our next trick (this time more than just a trick: a deep utilization of non-trivial properties of the thermodynamical limit $N \to \infty$), resulting again in a linearization in the exponent, is to leave $q_{\rho\sigma}$ formally a free variable, however include its definition as a delta function for which a suitable integral representation can be found.

Thus, the first exponential will appear as a product of integrals for each $\mu > s$:

$$\int \prod_{\rho\sigma(\rho<\sigma)} dq_{\rho\sigma}\delta\Big(q_{\rho\sigma} - (1-\delta_{\rho\sigma})\frac{1}{N}\sum_i S_i^\rho S_i^\sigma\Big) \times \exp\Big\{-\frac{\beta N}{2}\big[(1-\beta)\sum_\rho m_{\mu\rho}^2 - \beta\sum_{\rho\neq\sigma} m_{\mu\rho}m_{\mu\sigma}q_{\rho\sigma}\big]\Big\}. \quad (A.14)$$

The integrals have a characteristic structure: they represent something of the form

$$e^{Ng(Q)} = \int dq e^{Ng(q)}\delta(q-Q), \quad (A.15)$$

where Q is a short-hand notation for the right-hand side of (A.12). The integral representation we are looking for would take advantage of the presence of N in the exponent that makes all extrema (in complex variables: *saddle points*) of the exponential function very sharp, and approximate (A.15) by the double integral

$$\approx \int dr \int dq e^{N\left(g(q)-r(q-Q)\right)} \quad (A.16)$$

the integrand of which has sharp features in both variables. The r in the exponent is a Lagrange multiplier expected to push the sharp double maximum of the integrand (if it exists) to $q = Q$.

Let us see how this happens.

(i) Take the q-integral first. For a fixed r, the nonlinear growth of $g(q)$ is counterbalanced by $-rq$ at the point of maximum $q^*(r)$ where $g'(q^*(r)) = r$.

(ii) The exponent at the maximum, $N\big[g(q^*(r)) - rq^*(r) + Qr\big]$, still depends on r. The nonlinear decrease of the first two terms in the square bracket is counterbalanced by the growth of Qr at a point r^* of maximum where – just check! – $q^*(r^*) = Q$. A very sharp (for $N \gg 1$) maximum of the two-variable integrand at $r = r^*$, $q = q^*(r^*)$ has just the value $e^{Ng(Q)}$.

We are practically there; what we still need is the familiar *saddle-point approximation*: expand the *exponent* about the point of maximum to second-order terms; since everything is multiplied by N, these bring the exponential function down to nothing and no more terms are needed. Then integrating this sharp peak, the whole integral turns out to be just $e^{Ng(Q)+\varphi(N)}$, where $\varphi(N)$ is a logarithmically growing function of N. Neglecting it with respect to the linear $Ng(Q)$, what is called *saddle-point accuracy*, we obtain $e^{Ng(Q)}$ for the integral itself, as required.

This seems a long way round, but the result pays. We evaluate (A.13) by introducing a different Lagrange multiplier $r_{\rho\sigma}\alpha\beta^2/2$ for each condition $\delta(q_{\rho\sigma} - \ldots)$, to obtain

$$\begin{aligned}\big\langle\big\langle Z^n\big\rangle\big\rangle =& e^{-\frac{\beta}{2}pn}\,\mathrm{Tr}_{S^\rho}\int\prod_{\mu\rho}\frac{dm_{\mu\rho}}{\sqrt{2\pi}}\int\prod_{\rho\sigma(\rho<\sigma)}dr_{\rho\sigma}dq_{\rho\sigma}\\ \exp\Big\{&-\frac{\beta N}{2}\sum_{\mu>s}\big[(1-\beta)\sum_{\rho}m_{\mu\rho}^2-\beta\sum_{\rho\neq\sigma}m_{\mu\rho}m_{\mu\sigma}q_{\rho\sigma}\big]\\ &-\frac{N\alpha\beta^2}{2}\sum_{\rho\neq\sigma}r_{\rho\sigma}(q_{\rho\sigma}-\frac{1}{N}\sum_i S_i^\rho S_i^\sigma)\Big\}\int\prod_{\nu\rho}\frac{dm_{\nu\rho}}{\sqrt{2\pi}}\\ &\big\langle\big\langle\exp\big\{N\beta\big[-\frac{1}{2}\sum_{\nu\rho}^{(\nu\leq s)}m_{\nu\rho}^2+\sum_{\nu\rho}^{(\nu\leq s)}(m_{\nu\rho}+h_\nu)\frac{1}{N}\sum_i\xi_i^\nu S_i^\rho\big]\big\}\big\rangle\big\rangle_{\{\xi_i^\nu\}}\end{aligned}\qquad (A.17)$$

Now in all integrals we take the saddle-point values which identifies them as the order parameters learned in Section 4.3. Saddle points are where the respective partial derivatives of the exponent vanish:

(i) For the retrieved ("condensed") patterns $\nu \leq s$, $\frac{\partial}{\partial m_{\nu\rho}} = 0$ gives $-m_{\nu\rho}\big\langle\big\langle Z^n\big\rangle\big\rangle + \ldots\ \mathrm{Tr}\int\big\langle\big\langle\ldots\frac{1}{N}\sum_i\xi_i^\nu S_i^\rho\big\rangle\big\rangle = 0$, *i.e.*

$$m_{\nu\rho} = \langle\langle \frac{1}{N} \sum_i \xi_i^\nu \langle S_i^\rho \rangle \rangle\rangle, \quad \nu = 1 \ldots s, \tag{A.18}$$

which is just the retrieval quality order parameter on pattern ν for the quenched thermal equilibrium of replica ρ.

(ii) In the same way from $\frac{\partial}{\partial r_{\rho\sigma}}$ one obtains the Edwards-Anderson order parameter:

$$q_{\rho\sigma} = \langle\langle \frac{1}{N} \sum_i \langle S_i^\rho \rangle \langle S_i^\sigma \rangle \rangle\rangle \tag{A.19}$$

that in the replica picture characterizes the correlations caused by the equality of quenched patterns between two otherwise thermally independent replicas.

(iii) $\frac{\partial}{\partial m_{\mu\rho}} = 0$ for the non-retrieved patterns gives $m_{\mu\rho} = N^{-1} \sum_i \xi_i^\mu \langle S_i^\rho \rangle$ with fixed (non-averaged) components of $\{\xi^{(\mu)}\}$; then $\frac{\partial}{\partial q_{\rho\sigma}} = 0$ identifies

$$r_{\rho\sigma} = \frac{1}{\alpha} \sum_{\mu=s+1}^{\alpha N} \langle\langle m_{\mu\rho} m_{\mu\sigma} \rangle\rangle \tag{A.20}$$

as the order parameter of random overlaps with the non-retrieved patterns.

We are left with two more non-trivial operations. Let us first exploit the quadratic expansion in the non-retrieved patterns, equation (A.10). For that reason, the integrals over $m_{\mu\rho}$ in (A.17) are just multi-dimensional Gaussian, easy to do by transforming to principal axes. To this end one has to find the n eigenvalues λ_α of the matrix

$$(1-\beta)\mathbf{I} - \beta\mathbf{q} = \begin{pmatrix} 1-\beta & -\beta q_{12} & \ldots & -\beta q_{1n} \\ -\beta q_{21} & 1-\beta & \ldots & -\beta q_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -\beta q_{n1} & -\beta q_{n2} & \ldots & 1-\beta \end{pmatrix} \tag{A.21}$$

from which the Gaussian integral is obtained as

$$\begin{aligned} &\int \prod_{\mu\alpha} \frac{dy_\alpha}{\sqrt{2\pi}} e^{-\frac{1}{2}\sum_\alpha \lambda_\alpha y_\alpha^2} \\ &= \prod_{\mu\alpha} \frac{1}{\sqrt{\lambda_\alpha}} = \prod_\mu e^{-\frac{1}{2}\sum_\alpha \ln \lambda_\alpha} = \mathrm{Tr} \ln[(1-\beta)\mathbf{I} - \beta\mathbf{q}] \,. \end{aligned} \tag{A.22}$$

The evaluation of the eigenvalues λ_α is easy for the replica symmetric case (see below).

The other important and nontrivial step refers to averaging the line with $\langle\langle \ldots \rangle\rangle$ in (A.17) over the retrieved patterns. Since the number s of retrieved patterns is *finite* while $N \to \infty$, the overall noise caused by these patterns is averaged out just by doing the operation $N^{-1}\sum_i$ over the large network, to leave very little fluctuation already in the exponent (this property is called *self-averaging*; the reason for all our troubles in this Appendix is that it cannot be done if noise from all αN patterns has to be included). Consequently, one can safely approximate

$$\langle\langle \mathrm{Tr}_{S^\rho} \exp \ldots \rangle\rangle = \exp\langle\langle \ln \mathrm{Tr}_{S^\rho} \exp \ldots \rangle\rangle. \qquad (A.23)$$

Our final (yes!) expression for the averaged partition function is

$$\langle\langle Z^n \rangle\rangle = e^{-\frac{\beta}{2}pn} \int \prod_{\nu\rho}^{[1,s]} \frac{dm_{\nu\rho}}{\sqrt{2\pi}} \int \prod_{\rho\sigma} dr_{\rho\sigma} dq_{\rho\sigma}$$
$$\times \exp N \Big\{ -\frac{1}{2}\beta \sum_{\nu\rho} m_{\nu\rho}^2 - \frac{\alpha}{2} \mathrm{Tr} \ln[(1-\beta)\mathbf{I} - \beta\mathbf{q}] - \frac{\alpha\beta^2}{2} \sum_{\rho\neq\sigma} r_{\rho\sigma} q_{\rho\sigma}$$
$$+ \langle\langle \ln \mathrm{Tr}_{S^\rho} e^{\beta H_\xi} \rangle\rangle_{\xi_i^\nu} \Big\}, \qquad (A.24)$$

where

$$H_\xi = \frac{\alpha\beta}{2} \sum_{\rho\neq\sigma} r_{\rho\sigma} S^\rho S^\sigma + \sum_{\nu\rho} (m_{\nu\rho} + h^\nu) \xi^\nu S^\rho \qquad (A.25)$$

can be regarded as an effective 1-spin Hamiltonian for a fixed choice of the retrieved patterns ξ^ν.

What remains is three last operations which are rather straightforward, therefore we do not go into the details:

(i) calculate the free energy f_n as a function of the variational parameters $m_{\nu\rho}$, $q_{\rho\sigma}$ and $r_{\rho\sigma}$ and the number n of replicas; for those who still follow the calculation: this is

$$f_n = \frac{\alpha}{2} + \frac{\alpha}{2\beta n}\mathrm{Tr}\ln[(1-\beta)\mathbf{I} - \beta\mathbf{q}] + \frac{1}{2n}\sum_{\nu\rho}(m_{\nu\rho})^2 + \frac{\alpha\beta}{2n}\sum_{\rho\neq\sigma} r_{\rho\sigma}q_{\rho\sigma} - \frac{1}{n\beta}\langle\langle \ln \mathrm{Tr}_S e^{\beta H_\xi}\rangle\rangle_\xi ; \qquad (A.26)$$

(ii) as a first approximation, allow only a little subset of the variational parameters in which all replicas or replica pairs respectively are assigned the same parameter values, which is called *replica symmetry*:

$$m_{\nu\rho} = m^\nu; \quad q_{\rho\sigma} = q; \quad r_{\rho\sigma} = r;$$

then f remains to be minimized with respect to these three parameters only; besides, $\mathrm{Tr}\ln[(1-\beta)\mathbf{I} - \beta\mathbf{q}]$ is now easy to calculate since the matrix (A.21) has simple eigenvectors: one of all vector components equal; $n-1$ of components of zero sum;

(iii) take the limit $n \to 0$. On doing this, we have to calculate $\mathrm{Tr}_{S^\rho} e^{\beta H_\xi}$ which is not simple because H_ξ is quadratic in the spins, but we know the remedy: one more Gaussian transform of the kind (A.7)–(A.8) has to be done, introducing one more Gaussian-distributed multiplicative noise denoted by z.

All this being done, one obtains the replica-symmetric free energy

$$f = \frac{1}{2}\alpha + \frac{1}{2}\sum_\nu (m^\nu)^2 + \frac{\alpha}{2\beta}\left[\ln(1-\beta(1-q)) - \frac{\beta q}{1-\beta(1-q)}\right] + \frac{\alpha\beta}{2}r(1-q) - \frac{1}{\beta}\left\langle\left\langle \ln 2\cosh\beta[\sqrt{\alpha r}z + \sum_\nu (m^\nu + h^\nu)\xi^\nu]\right\rangle\right\rangle. \qquad (A.27)$$

Here we can recognize $\sqrt{\alpha r}z + \sum_\nu (m^\nu + h^\nu)\xi^{nu}$ as the local field acting on a single spin, $\sqrt{\alpha r}z$ being its random, spin-glass-like part: the noise from random overlaps with non-retrieved patterns. The brackets

$$\langle\langle \ldots\rangle\rangle \equiv \int \frac{dz}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} \langle\langle \ldots\rangle\rangle_{\{\xi^\nu\}} \qquad (A.28)$$

in the final formulas represent a double averaging: over the retrieved patterns $\xi^\nu = \pm 1$, and over the Gaussian multiplicative noise z.

Taking the extremum of f with respect to the three replica-symmetric variational parameters m^ν, q and r, one obtains the three mean-field equations (4.23), (4.28) and (4.31) of the text (we notice that to obtain (4.28) in the standard form, a partial integration in z has to be done).

Although replica symmetry has been introduced as an approximation in which the distributions of parameters over replicas and replica pairs respectively are roughly characterized by averages, the solution of the full, unrestricted variational problem, allowing *replica symmetry breaking,* may turn out to be replica symmetric in some range of α and β. Easier than starting from full generality, this can be checked by allowing just a little more than three variational parameters and see whether their minimization gives back the fully symmetric solution or not. Such a calculation has been done (Crisanti *et al.* 1986) with the result that effects of replica symmetry breaking for the plain Hopfield model are restricted to a small part of the phase diagram and are not very significant. This could not be guessed however without doing the calculation, and there is no guarantee of the same good luck for modified models.

APPENDIX B

DYNAMICS OF A TWO-PATTERN NETWORK

In this appendix we follow the treatment of Pázmándi and Geszti (1989). This paper generalizes the one-pattern model (Krauth *et al.* 1988) to the case of two patterns ξ_i^1 and ξ_i^2 (two-pattern network: TPN) of given stabilities Δ_1 and Δ_2:

$$\sum_j \xi_i^{1,2} J_{ij} \xi_j^{1,2} = \Delta_{1,2} \tag{B.1}$$

and a given value of the coupling symmetry parameter η (equation (7.14)), all independent of the neuron i ($i = 1 \ldots N$), where $J_{ii} = 0$ and the clipped connection strengths $J_{ij} = \pm N^{-1/2}$ for $i \neq j$ are satisfying the above constraints but otherwise random. As a further simplification, we assume that the two stored patterns are orthogonal.

At time t the configuration of the model is characterized by a vector $\mathbf{S}^t = \{S_i^t\}$. We start by generating initial configurations out of a distribution

$$P_1(\mathbf{S}^0) = \prod_{i=1}^{N} \frac{1 + S_i^0 \sum_{\mu=1,2} \xi_i^\mu m_\mu}{2}, \tag{B.2}$$

specified by the initial values m_μ of the overlaps (7.1) with the stored patterns.

Let the configurations evolve by parallel dynamics at temperature T. We investigate the change of some average properties, like the overlaps themselves.

Parallel dynamics of the model (Section 2.3) proceeds in independent simultaneous single-spin flips. It is advantageous to rewrite equation (2.7) as a closed expression for the probability of a given value S_i^t at time t:

$$W(S_i^t \mid \mathbf{S}^{t-1}) = \frac{1 + S_i^t f(z_i^{t-1})}{2} \tag{B.3}$$

with $f(x) = \tanh(\beta x)$, $\beta = 1/T$ and

$$z_i^t = \sum_j J_{ij} S_j^t. \tag{B.4}$$

This defines a Markovian evolution in which the probability distribution $P_1(\mathbf{S}^t)$ is fully determined by $P_1(\mathbf{S}^{t-1})$ through the transition probability

$$W(\mathbf{S}^t \mid \mathbf{S}^{t-1}) = \prod_{i=1}^{N} W(S_i^t \mid \mathbf{S}^{t-1}). \tag{B.5}$$

In order to calculate the time evolution of the overlaps (7.1), where

$$\begin{aligned} \langle S_i^t \rangle &= Tr_{\mathbf{S}^t} S_i^t P_1(\mathbf{S}^t) \\ &= Tr_{S_i^t} S_i^t P_1(S_i^t), \end{aligned} \tag{B.6}$$

we need an approximate expression for

$$P_1(S_i^t) = Tr_{\mathbf{S}^{t-1},\mathbf{S}^{t-2},\ldots,\mathbf{S}^0} W(S_i^t \mid \mathbf{S}^{t-1}) W(\mathbf{S}^{t-1} \mid \mathbf{S}^{t-2}) \ldots W(\mathbf{S}^1 \mid \mathbf{S}^0) P_1(\mathbf{S}^0). \tag{B.7}$$

As we shall see below, dynamics is dominated by the parameters Δ_1, Δ_2 and η. These parameters are site-independent. Therefore we expect $\langle S_i^t \rangle$ to depend on i only through the values of ξ_i^1 and ξ_i^2 at the same site i. Since these are binary variables, $\langle S_i^t \rangle$ can be written as a linear function of them. Moreover, this function is homogeneous because it is multiplied by -1 if so are both of its arguments. Then its two coefficients are determined by the orthogonality of the two patterns, which gives the important formula

$$\langle S_i^t \rangle = \sum_{\nu} \xi_i^{\nu} m_{\nu}(t). \tag{B.8}$$

that inverts equation (7.1). This is essentially an expression of self-averaging for m_μ, which – using (B.8) instead of (7.1) – can be evaluated by averaging $\xi_i^\mu \langle S_i^t \rangle$ over the patterns, instead of over sites:

$$\frac{1}{N} \sum_i \ldots \;\rightarrow\; \overline{(\ldots)}^{\xi}. \tag{B.9}$$

The same holds for all gauge-invariant quantities, but we emphasize that only for the restricted class of strict-stability models with site-independent stabilities.

Below we illustrate how this works for the first two time steps. For $t = 1$ a straightforward application of equations (7.1), (B.6), (B.7) and (B.3) gives

$$m_\nu(1) = \langle\langle \xi_i^\nu f(z_i^0)\rangle\rangle_{\xi,z} \tag{B.10}$$

where $\langle\langle \ldots\rangle\rangle_{\xi,z}$ means averaging over patterns and a distribution of the random sums z_i^t defined in equation (B.4). By the central limit theorem z_i^0 is a Gaussian random variable. Its mean value x_i^0 and squared dispersion D_0 can be calculated from equation (B.8):

$$x_i^0 = \sum_\mu \xi_i^\mu \Delta_\mu m_\mu, \tag{B.11}$$

and, since $\overline{(z_i^0)^2} = 1 + \sum_\mu (\Delta_\mu^2 - 1) m_\mu^2$,

$$D_0 = 1 - \sum_\mu m_\mu^2. \tag{B.12}$$

Then

$$m_\nu(1) = \int \frac{dy}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} \overline{\left[\xi_i^\nu f(x_i^0 + \sqrt{D_0}y)\right]}^{\xi}. \tag{B.13}$$

Let us turn to the next time step, $t = 2$. By equation (B.3), we have to average over the conditional distribution $P(z_i^1 \mid \mathbf{S}^0)$. Now the conditional mean value $\langle z_i^1\rangle_{\mathbf{S}^0} = \sum_j J_{ij}\langle S_j^1\rangle_{\mathbf{S}^0}$ is determined iteratively, by the first time step as calculated above. Here, however, due to the sum weighted by J_{ij}, an important coherent contribution arises. To see this, in

$$\langle S_j^1\rangle_{\mathbf{S}^0} = \langle\langle f(z_j^0) P(z_j^0 \mid \mathbf{S}^0)\rangle\rangle \tag{B.14}$$

let us single out from $z_j^0 = \sum_k J_{jk} S_k^0$ both its mean and the fluctuation of the term $k = i$:

$$z_j^0 = x_j^0 + J_{ji}(S_i^0 - \langle S_i^0\rangle) + \sqrt{D_0}y \tag{B.15}$$

(explicitely adding one fluctuation term gives a negligible change of D_0), and linearize $f(z_j^0)$ in the J_{ji} term which is small since $J_{ji} \propto N^{-1/2}$ whereas $z_j^0 = \mathcal{O}(1)$:

$$f(z_j^0) \approx f(x_j^0 + \sqrt{D_0}y) + J_{ji} f'(x_j^0 + \sqrt{D_0}y)(S_i^0 - \langle S_i^0 \rangle). \qquad (B.16)$$

Substituting into z_i^1 (see equation (B.4)), the j summation gives a contribution proportional to η (equation (1)), which is $\mathcal{O}(1)$, expressing the dominant influence of the initial value of a given spin S_i on its own mean value *two* time steps later, through the correlation between J_{ij} and J_{ji} expressed by $\eta \neq 0$.

The rest of the calculation is trivial and gives

$$m_\nu(2) = \langle\langle \xi_i^\nu Tr_{S_i^0} \frac{1 + S_i^0 \sum_\mu \xi_i^\mu m_\mu}{2} f(x_i^1 + \sqrt{D_1}y) \rangle\rangle, \qquad (B.17)$$

where

$$x_i^1 = \sum_\mu \xi_i^\mu \Delta_\mu m_\mu(1) + (S_i^0 - \sum_\mu \xi_i^\mu m_\mu) \eta V_{01}, \qquad (B.18)$$

$$D_1 = 1 - \sum_\mu m_\mu^2(1) \qquad (B.19)$$

(based on the estimate $\sum_{ik} J_{ij} J_{ik} J_{jk} = \mathcal{O}(N^{-1/2})$ which is not true *e.g.* for the fully connected Hopfield net: see Gardner *et al.* 1987), and

$$V_{01} = \langle\langle f'(x_i^0 + \sqrt{D_0}y) \rangle\rangle. \qquad (B.20)$$

For longer times ($t > 2$) we have to do some bookkeeping of correlations due to coherent terms mediated by the symmetry η, connecting always the same spin two time units apart. They appear in two different forms: explicitely for $t-2$, $t-4 \ldots$; implicitely as Gaussian noise cross-correlations for $t-1$, $t-3 \ldots$

The starting point is equation (B.7) in which a chain product apppears, each factor being a product itself according to (B.5). Out of all this, the factors $W(S_i^t \mid \mathbf{S}^{t-1})$, $W(S_i^{t-2} \mid \mathbf{S}^{t-3}) \ldots$ require special attention. From (B.3) we see that they depend on $\mathbf{S}^{t-1}, \mathbf{S}^{t-3} \ldots$ only through the combinations z_i^{t-1}, $z_i^{t-3} \ldots$, therefore instead of calculating $Tr_{\mathbf{S}^{t-1}, \mathbf{S}^{t-2} \ldots}$, we average over these scalar variables. Doing the rest of the traces, for the overlap (7.1) we obtain an expression of the form

$$m_\nu(t) = Tr_{S_i^{t-2}, S_i^{t-4}\ldots} \langle\langle \xi_i^\nu f(z_i^{t-1}) W(S_i^{t-2} \mid z_i^{t-3}) W(S_i^{t-4} \mid z_i^{t-5}) \ldots \rangle\rangle, \quad (B.21)$$

where the double average has to be interpreted as

$$\langle\langle \ldots \rangle\rangle = \int dz_i^{t-1} dz_i^{t-3} \ldots P_n(z_i^{t-1}, z_i^{t-3} \ldots \mid S_i^{t-2}, S_i^{t-4} \ldots) \overline{\left(\ldots\right)}^{\xi}. \quad (B.22)$$

Our next task is to obtain an explicite expression for the conditional probability distribution P_n.

Since $J_{ii} = 0$, none of the terms in $z_i^\tau = \sum_j J_{ij} S_j^\tau$ for a given time τ is immediately correlated with S_i^t for any time. Some harder content to this intuitive statement is that the evolution of any S_j^τ is influenced by that of many spins S_k^t among which only $k = j$ has a distinguished role, $k = i$ is just one out of many.

For this reason we expect it is reasonable to approximate the joint distribution of variables z_i^τ for different values of τ by a multi-dimensional Gaussian

$$P_n(z_i^{t-1}, z_i^{t-3} \ldots \mid S_i^{t-2}, S_i^{t-4} \ldots) \approx (2\pi)^{-n/2} \mid D \mid^{-1/2} \exp\Big\{-\frac{1}{2} \sum_{\tau,\tau'=t-1,t-3\ldots} D^{-1}_{\tau\tau'} \left[(z_i^\tau - x_i^\tau)(z_i^{\tau'} - x_i^{\tau'})\right]\Big\} \quad (B.23)$$

where

$$x_i^\tau = \sum_j J_{ij} \langle S_j^\tau \rangle_{(i)} \quad (B.24)$$

(here $\langle \ldots \rangle_{(i)}$ means that on calculating the average S_i^{t-2}, $S_i^{t-4} \ldots$ have fixed values), and

$$D_{\tau\tau'} = \sum_{jk} J_{ij} J_{ik} \left(\langle S_j^\tau S_k^{\tau'} \rangle - \langle S_j^\tau \rangle \langle S_k^{\tau'} \rangle\right) \quad (B.25)$$

in which the same restriction has not to be taken explicitely into account since its effect is negligible in the thermodynamical limit. Finally $\mid D \mid$ denotes the determinant of the matrix $D_{\tau\tau'}$.

The rest of our task is to determine the noise centers x_i^τ and the dispersions $D_{\tau\tau'}$ for $\tau, \tau' = t-1, t-3 \ldots$. It is in this step that the linearization trick mentioned above can be applied to obtain $\mathcal{O}(\eta)$ corrections of two kinds to the fully asymmetric case $\eta = 0$: shifts of x_i^τ, and non-diagonal elements of $D_{\tau,\tau'}$ which would not appear at all without the correlations induced by the symmetry of coupling coefficients.

For the details of the calculation the Reader is referred to the original paper (Pázmándi and Geszti 1989). The results are:

$$x_i^{t-1} = \sum_\mu \xi_i^\mu \Delta_\mu m_\mu^{t-1} + \eta \sum_{\tau=t-2,t-4\ldots} (S_i^\tau - \sum_\mu \xi_i^\mu m_\mu^\tau) V_{\tau,t-1} \qquad (B.26)$$

(and analogously for x_i^{t-3}, *etc.*), where

$$V_{\tau,t-1} = Tr_{S_j^{t-1}, S_j^{t-3}\ldots} \langle\langle \frac{1}{2} S_j^{t-1} S_j^\tau f'(z_j^\tau) \prod_{\tau'=t-1,t-3\ldots}^{\tau' \neq \tau} \frac{1 + S_j^{\tau'} f(z_j^{\tau'-1})}{2} \rangle\rangle, \quad (B.27)$$

and

$$D_{\tau\tau'} = q_{\tau\tau'} - \sum_\mu m_\mu(\tau) m_\mu(\tau') \qquad (B.28)$$

where

$$q_{\tau\tau'} = \langle\langle Tr_{S_j^\tau S_j^{\tau'}} S_j^\tau S_j^{\tau'} P_2(S_j^\tau, S_j^{\tau'}) \rangle\rangle \qquad (B.29)$$

and P_2 is a double-time distribution function P_2 defined by taking the trace of the full distribution over everything but S_j^τ and $S_j^{\tau'}$. In the Gaussian approximation *e.g.* for $\tau > \tau'$ it gives

$$P_2(S_j^\tau, S_j^{\tau'}) = Tr_{S_j^{\tau-2}, S_j^{\tau-4}\ldots S_j^{\tau'+2}, S_j^{\tau'-2}\ldots} \int dz_j^{\tau-1} dz_j^{\tau-3} \ldots$$
$$\left[\frac{1 + S_j^\tau f(z_j^{\tau-1})}{2} \frac{1 + S_j^{\tau-2} f(z_j^{\tau-3})}{2} \ldots \right] P(z_i^{\tau-1}, z_j^{\tau-3} \ldots \mid S_j^{\tau-2} S_j^{\tau-4} \ldots), \qquad (B.30)$$

where P is a Gaussian with parameters already iteratively determined in the earlier time steps. In particular, the diagonal elements are

$$D_{\tau\tau} = 1 - \sum_{\mu} m_{\mu}^2(\tau). \tag{B.31}$$

To illustrate the above formalism, we give the formulas pertinent to $t = 3$, to be added to the list of equations (B.17) to (B.20):

$$m_{\nu}(3) = \langle\langle \xi^{\nu} Tr_{S^1} \frac{1 + S^1 f(z^0)}{2} f(z^2) \rangle\rangle, \tag{B.32}$$

$$x_2 = \sum_{\mu} \xi^{\mu} \Delta_{\mu} m_{\mu}(2) + \big(S^1 - \sum_{\mu} \xi^{\mu} m_{\mu}(1)\big) \eta V_{12}, \tag{B.33}$$

$$V_{12} = \langle\langle Tr_{S^0} \frac{1 + S^0 \sum_{\mu} \xi^{\mu} m_{\mu}}{2} f'(z^1) \rangle\rangle, \tag{B.34}$$

$$q_{02} = \langle\langle Tr_{S^0} \frac{S^0 + \sum_{\mu} \xi^{\mu} m_{\mu}}{2} f(z^1) \rangle\rangle. \tag{B.35}$$

Treating longer times is straightforward, apart from the need of calculating averages over Gaussian distributions of more and more dimensions.

The vanishing of the non-diagonal elements of $D_{\tau\tau'}$ (the connected correlation functions between different time values) in the zero-symmetry case $\eta = 0$ is not explicite in the formulas obtained. It can be proven in two steps: one demonstrates (i) $q_{0\tau} = \sum_{\mu} m_{\mu} m_{\mu}(\tau)$ *i.e.* $D_{0\tau} = 0 \ \forall \tau \neq 0$; (ii) the vanishing of non-diagonal elements propagates from $(\tau - 1, \tau' - 1)$ to (τ, τ'), which proves the assertion by induction.

In view of this simplification for $\eta = 0$ one has a one-step iteration for the overlaps. Moreover, the requirement of site-independent stabilities can be relaxed to that of fluctuating stabilities Δ_{i1}, Δ_{i2} of the same distribution $P(\Delta_1, \Delta_2)$ on all sites. Then the iteration is

$$m_{\nu}(t+1) = \int d\Delta_1 d\Delta_2 P(\Delta_1 \Delta_2) \langle\langle \xi^{\nu} f\Big(\sum_{\mu} \xi^{\mu} \Delta_{\mu} m_{\mu}(t) + \sqrt{D_0} z\Big) \rangle\rangle, \tag{B.36}$$

where z is a Gaussian variable of zero mean and unit dispersion, and

$$D_0 = 1 - \sum_{\mu} m_{\mu}^2(t) \tag{B.37}$$

under the normalization

$$\sum_{j} J_{ij}^2 = 1. \tag{B.38}$$

Equation (B.37) expresses an important property of strict-stability networks: perfect retrieval of a pattern allows no fluctuation at all; close to perfect retrieval allows but a much reduced level of noise.

For the two-pattern network model defined by (B.1) Δ_1 and Δ_2 are sharply determined and $\sqrt{D_0}z$ is the only noise disturbing the pattern retrieval. Then for $\eta = 0$

$$m_{\nu}(t+1) = \langle\langle \xi^{\nu} f\Big(\sum_{\mu} \xi^{\mu}\Delta_{\mu} m_{\mu}(t) + \sqrt{D}z\Big)\rangle\rangle. \tag{B.39}$$

with

$$D = D_0. \tag{B.40}$$

In a Hopfield-type network with Hebbian learning the noise reduction effect described by (B.37) is opposed by the Gaussian-distributed fluctuations of the stabilities themselves, which add noise terms of amplitudes proportional to $m_1(t)$ and $m_2(t)$. For the case of asymmetric dilution this just compensates the noise reduction terms in equation (B.37) and for the evolution of $m_{\nu}(t)$ we obtain an equation of the same form as (B.39), however, with D_0 replaced by

$$D = 1. \tag{B.41}$$

This enhanced noise level is the main reason why patterns in Hopfield-type networks have to be kept at a rather high level of stability to assure any retrieval.

REFERENCES

Abbott L. F. and Kepler T. B. (1988) *Optimal learning in neural network memories,* preprint: Brandeis University BRX-TH-255

Abeles M. (1982) *Local Cortical Circuits* (Springer, Berlin)

Abramowitz M. and Stegun I. A. (1972) *Handbook of Mathematical Functions* (Dover Publ., New York)

Ackley D. H., Hinton G. E. and Sejnowski T. J. (1985) *Cognitive Science* **9,** 147

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

Amit D. J. (1988) *Proc. Natl. Acad. Sci. USA* **85,** 2141

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

Amit D. J., Gutfreund H. and Sompolinsky H. (1987) *Phys. Rev. A* **35,** 2293

Amit D. J., Gutfreund H. and Sompolinsky H. (1987) *Ann. Phys. (NY)* **173,** 30

Amit D. J., Sagi D. and Usher M. (1989) *Architecture of Attractor Neural Networks Performing Cognitive Fast Scanning:* preprint

Baldi P. and Venkatesh S. S. (1987) *Phys. Rev. Lett.* **58,** 913

Bialek W. and Zee A. (1987) *Phys. Rev. Lett.* **58,** 741

Bialek W. and Zee A. (1988) *Phys. Rev. Lett.* **61,** 1512

Bruce A. D., Gardner E. J. and Wallace D. J. (1987) *J. Phys. A: Math. Gen.* **20,** 2909

Buhmann J. and Schulten K. (1987) *Europhys. Lett.* **4,** 1205

Buhmann J., Divko R. and Schulten K. (1989) *Phys. Rev. A* **39,** 2689

Carnevali P. and Patarnello S. (1987) *Europhys. Lett.* **4,** 1199

Clark J. W., Rafelski J. and Winston J. V. (1985) *Physics Repts.* **123,** 215

Cortes C., Krogh A. and Hertz J. A. (1987) *J. Phys. A: Math. Gen.* **20,** 4449

Cottrell M. and Fort J. C. (1986) *Biol. Cybern.* **53,** 405

Cottrell M. and Fort J. C. (1987) *Ann. Inst. Henri Poincaré* **23,** 1

Cover T. M. (1965) *IEEE Trans.* **EC-14,** 326

Crick F. and Mitchison G. (1983) *Nature* **304,** 111

Crisanti A., Amit D. J. and Gutfreund H. (1986) *Europhys. Lett.* **2,** 337

Crisanti A. and Sompolinsky H. (1988) *Phys. Rev. A* **37,** 4865

Czakó F. (1989) *unpublished*

Dehaene S., Changeux J. P. and Nadal J. P. (1987) *Proc. Natl. Acad. Sci. USA* **84,** 2727

Denker J., Schwartz D. Wittner B., Solla S., Howard R., Jackel L. and Hopfield J. (1987) *Complex Systems* **1,** 877

Derrida B., Gardner E. and Zippelius A. (1987) *Europhysics Letters* **4,** 167

Derrida B. and Nadal J. P. (1987) *J. Stat. Phys.* **49,** 993

Diederich S. and Opper M. (1987) *Phys. Rev. Lett.* **58,** 949

Dotsenko V. S. (1985) *J. Phys. C* **18,** L1017

Dotsenko V. S. (1986) *Physica* **140A,** 410

Dotsenko V. S. (1988) *J. Phys. A: Math. Gen.* **21,** L783

Dotsenko V. S. and Feigelman M. V. (1989) *"Relearning" algorithm for the Hopfield model of neural networks,* preprint

Feigelman M. V. and Ioffe L. B. (1986) *Europhys. Lett.* **1,** 197

Feigelman M. V. and Ioffe L. B., (1987) *Int. J. Mod. Phys.* **B.1,** 51

Fontanari J. F. and Köberle R. (1988a) *J. Phys. A: Math. Gen.* **21,** L259

Fontanari J. F. and Köberle R. (1988b) *J. Phys. A: Math. Gen.* **21,** L667

Forrest B. M. (1988) *J. Phys. A: Math. Gen.* **21, 245**

Gallant S. I. (1987) *Optimal linear discriminants,* preprint: Northeaster University (Boston), to appear in: *Proc. Int. Conf. on Pattern Recognition, Paris*

Gardiner C. W. *A Handbook of Stochastic Methods*, 2nd ed. (Springer, New York)

Gardner E. (1988) *J. Phys. A: Math. Gen.* **21,** 257

Gardner E. and Derrida B. (1988) *J. Phys. A: Math. Gen.* **21,** 271

Gardner E., Derrida B. and Mottishaw P. (1987) *J. Physique* **48,** 741

Gardner E., Stroud N. and Wallace D. J. (1989) *J. Phys. A: Math. Gen.* **22,** 2019

Geszti T. (1986) *Physics Lett.* **114A,** 334

Geszti T. and Pázmándi F. (1987) *J. Phys. A: Math. Gen.* **20,** L1299

Geszti T. and Pázmándi F. (1989) *Phys. Scripta* **T25,** 152

Geszti T. and Csabai I. (1989) *Dynamics of the free and forced Kohonen map*: preprint

Grossberg S. (1976) *Biol. Cybern.* **21,** 145; **23,** 121

Grossman T., Meir R. and Domany E. (1988) *Complex systems* **2,** 555

Gutfreund H. (1988) *Phys. Rev. A* **37** 570

Gutfreund H. and Mézard M. (1988) *Phys. Rev. Lett.* **61,** 235

Gutfreund H., Reger J. D. and Young A. P. (1988) *J. Phys. A: Math. Gen.* **21,** 2775

Guyon I., Personnaz L., Nadal J. P. and Dreyfus G. (1988) *Phys. Rev. A* **38,** 6365

Hebb D. O. (1949) *The Organization of Behavior* (Wiley, New York)

Hecht-Nielsen R. (1988) *IEEE Spectrum* March number, p.36

van Hemmen J. L. (1986) *Phys. Rev. A* **34,** 3435

van Hemmen J. L., Keller G. and Kühn R. (1988) *Europhys. Lett.* **5,** 663

Hertz J. A., Grinstein G. and Solla S. A. (1987) in: *Heidelberg Colloquium on Glassy Dynamics* (ed. by J. L. van Hemmen and I. Morgenstern: Lecture Notes in Physics Vol. 275, Berlin: Springer) 538

Hertz J. A. (1988) *Unpublished*

Hertz J. A., Thorbergson G. I. and Krogh A. (1989a) *Phys. Scripta* **T25,** 149

Hertz J. A., Krogh A. and Thorbergsson G. I. (1989b) *J. Phys. A: Math. Gen.* **22,** 2133

Hertz J. A. and Palmer R. G. (1989) *Neural Computation* (to appear: Addison and Wesley)

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

Hopfield J. J. (1984) *Proc. Natl. Acad. Sci. USA,* **81,** 3088

Hopfield J. J., Feinstein D. I. and Palmer R. G. (1983) *Nature* **304,** 158

Hopfield J. J. and Tank D. W. (1985) *Biol. Cybern.* **52,** 141

Hopfield J. J. (1987) *Proc. Natl. Acad. Sci. USA* **84,** 8429

Horn D. and Usher M. (1988) *J. de Physique* **49,** 389

Horn D. and Usher M. (1989) *Phys. Rev. A* **40,** 1036

Kandel E. R. (1979) *Sci. Am.* **241,** 60

Kanter I. and Sompolinsky H. (1987a) *Phys. Rev. A* **35,** 380

Kanter I. and Sompolinsky H. (1987b) *Phys. Rev. Lett.* **58,** 164

Kantz H. and Grassberger P. (1985) *Physica* **17D,** 75

Kepler T. B. and Abbott L. F. (1988) *J. Physique* **49,** 1657

Kinzel W. (1985) *Zeitschr. Phys. B* **60,** 205

Kinzel W. (1987) in: *Heidelberg Colloquium on Glassy Dynamics* (ed. J. L. van Hemmen and I. Morgenstern; Lecture Notes in Physics Vol. 275, Springer, Berlin) 529

Kinzel W. and Opper M. (1989) in: *Physics of Neural Networks*, (ed. J. L. van Hemmen, E. Domany and K. Schulten; to appear: Springer, Berlin)

Kirkpatrick S. and Sherrington D. (1978) *Phys. Rev. B* **17,** 4384

Kirkpatrick S., Gelatt C. D. and Vecchi M. P. (1983) *Science* **220,** 671

Kleinfeld D. (1986) *Proc. Natl. Acad. Sci. (U.S.A.)* **83,** 9469

Knudsen E. I., du Lac S. and Esterly S. D. (1987) *Annual Review of Neuroscience*, Vol. **10**

Kohonen T. and Ruohonen M. (1973) *IEEE Trans. Comput.* **C-22,** 701

Kohonen T. (1982) *Biol. Cybern.* **44,** 135

Kohonen T. (1988) *Self-Organization and Associative Memory*, 2nd Edition (Springer, Berlin)

Kondor I. (1988) *private communication*

Krauth W. and Mézard M. (1987) *J. Phys. A: Math. Gen.* **20,** L745

Krauth W., Nadal J.-P. and Mézard M. 1988 *J. Phys. A: Math. Gen.* **21,** 2995

Krauth W. and Opper M. (1989) *J. Phys. A: Math. Gen.* **22,** L519

Krauth W. and Mézard M. (1989) *J. de Physique* **50,** 3057

Kree R. and Zippelius A. (1987) *Phys. Rev. A* **36,** 4421

Kree R. and Zippelius A. (1988) *J. Phys. A: Math. Gen.* **21,** L813

Krogh A. and Hertz J. A. (1988) *J. Phys. A: Math. Gen.* **21,** 2211

Kühn R., van Hemmen J. L. and Riedel U. (1989) it J. Phys. A: Math. Gen. **22,** 3123

LeCun Y. (1985) *in: Cognitiva (CESTA-AFCET Ed.)* p. 599

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

Little W. A. and Shaw G. L. (1978) *Math. Biosci.* **39,** 281

von der Malsburg C. and Bienenstock E. (1987) *Europhys. Lett.* **3,** 1243; **4,** 121

McCulloch W. S. and Pitts W. (1943) *Bull Math. Biophys.* **5,** 115

Mermin N. D. (1979) *Rev. Mod. Phys.* **51,** 591

Mézard M. (1989) *J. Phys. A: Math. Gen.* **22,** 2181

Mézard M., Nadal J. P. and Toulouse G. (1986) *J. Physique* **47,** 1457

Mézard M. and Nadal J. P. (1989) *J. Phys. A: Math. Gen.* **22,** 2191

Mézard M., Parisi G. and Virasoro M. A. (1988) *Spin Glass Theory and Beyond* (World Scientific, Singapore)

Minsky M. L. and Papert S. *Perceptrons*, MIT Press, Cambridge Ma. 1969 and 1988

Mori Y., Davis P. and Nara S. (1989) *J. Phys. A: Math. Gen.* **22,** L525

Nadal J. P., Toulouse G., Changeux J. P. and Dehaene S. (1986) *Europhysics Lett.* **1,** 535

Nadal J. P. (1989) *Study of a growth algorithm for a feedforward network,* preprint: L.P.S.E.N.S., submitted to *International Journal of Neural Systems*

Nadal J. P. and Toulouse G. (1989) *Information storage in sparsely coded memory nets,* preprint, to appear in: *Network, Compuation in Neural Systems*

Németh R. and Geszti T. (1988) *Acta Phys. Hung.* **64,** 59

Opper M. (1987) *Europhys. Lett.* **8,** 389

Opper M. (1988) *Phys. Rev. A* **38,** 3824

Opper M., Kleinz J., Köhler, H. and Kinzel, W. (1989) *J. Phys. A: Math. Gen.* **22,** L407

Parga N. and Virasoro M. A. (1986) *J. Physique* **47,** 1857

Parisi G. (1986a) *J. Phys. A: Math. Gen.* **19,** L617

Parisi G. (1986b) *J. Phys. A: Math. Gen.* **19,** L675

Parker D. B. (1985) *MIT Tech. Rept.* **TR-47**

Pázmándi F. (1988) *unpublished*

Pázmándi F. and Geszti T. (1989) *Relative stability in the dynamics of a two-pattern neural net,* to appear: *J. Phys. A: Math. Gen.* **22,** issue 22.

Pellionisz A. J. (1988) in: *Computer simulation in brain science* (ed. by Cotterill R. M. J., Cambridge University Press) p. 44

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

Peretto P. and Niez J. J. (1986a) *Biol. Cybern.* **54,** 53

Peretto P. and Niez J. J. (1986b) in: *Disordered Systems and Biological Organization* (ed. by Bienenstock E., Fogelman F. and Weisbuch G.: Springer, Berlin) p. 171

Peretto P. (1988a) *J. Physique* **49,** 711

Peretto P. (1988b) *Neural Networks* **1,** 309

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

Personnaz L., Guyon I. and Dreyfus G. (1985) *J. Physique Lett.* **46,** L359

Personnaz L., Guyon I. and Dreyfus G. (1986) *Phys. Rev. A* **34,** 4217

Personnaz L., Guyon I. and Dreyfus G. (1987) *Europhys. Lett.* **4,** 863

Peterson C. and Anderson J. R. (1987) *Complex Systems* **1,** 995

Poggio T. and Reichardt W. (1976) *Quart. Rev. Biophys.* **9,** 377

Pöppel G. and Krey U. (1987) *Europhys. Lett.* **4,** 979

Psaltis D., Park C. H. and Hong J. (1988) *Neural Networks* **1,** 149

Rammal R., Toulouse G. and Virasoro M. A. (1987) *Rev. Mod. Phys.* **58,** 765

Riedel U., Kühn R. and van Hemmen, J. L. (1988) *Phys. Rev. A* **38,** 1105

Rieger H., Schreckenberg M. and Zittartz J. (1988) *J. Phys. A: Math. Gen.* **21,** L263

Ritter H. and Schulten K. (1986) *Biol. Cybern.* **54,** 99

Ritter H. and Schulten K. (1988) *Biol. Cybern.* **60,** 59

Ritter H., Martinetz T. M. and Schulten K. (1989) *Neural Networks* **2,** 159

Rosenblatt F (1962) *Principles of Neurodynamics* (Spartan, New York)

Ruján P. and Marchand M. (1988) *A geometric approach to learning in neural networks,* Preprint: IFK Jülich

Rumelhart D. E., Hinton G. E. and Williams R. J. (1986) *Nature* **323,** 533

Rumelhart D. E. and McClelland J. L. (1986) *Parallel Distributed Processing* (MIT Press, Cambridge MA.)

Shinomoto S. (1987) *Biol. Cybern.* **57,** 197

Skarda C. A. and Freeman W. J. (1986) *Behavioral and Brain Sciences* (Cambridge University Press)

Solla S. A., Levin E. and Fleisher M. (1988) *Complex Systems* **2,** 625

Sompolinsky H. (1986) *Phys. Rev. A* **34,** 2571

Sompolinsky H. (1987) in: *Heidelberg Colloquium on Glassy Dynamics* (ed. J. L. van Hemmen and I. Morgenstern; Lecture Notes in Physics Vol. 275, Springer, Berlin) 485

Sompolinsky H. and Kanter I. (1986) *Phys. Rev. Lett.* **57,** 2861

Sourlas N. (1988) *Europhys. Lett.* **7,** 749

Sutton J. P., Beis J. S. and Trainor L. E. H. (1988) *J. Phys. A: Math. Gen.* **21,** 4443

Toulouse G. (1977) *Comm. in Physics* **2,** 115

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

Treves A. and Amit D. J. (1988) *J. Phys. A: Math. Gen.* **21,** 3155

Treves A. and Amit D. J. (1989) *J. Phys. A: Math. Gen.* **22,** 2205

Tsodyks M. V. and Feigelman, M. V. (1988) *Europhys. Lett.* **6,** 101

Tsodyks M. V. (1988) *Europhys. Lett.* **7,** 203

Venkatesh S. (1986) *Proc. Conf. on Neural Networks for Computing, Snowbird, Utah (AIP Conference Series* **151**) ed. J. S. Denker (AIP, New York) p. 440

Weisbuch G. and Fogelman-Soulié F. (1985) *J. Physique Lett.* **46,** L623

Weiss G. H. (1967) *Adv. Chem. Phys.* **13,** 1

Widrow G. and Hoff M. E. (1960) *IRE Western Electronic Show and Convention, Convention Record,* **Part 4,** p. 96

Willshaw D. J., Buneman O. P. and Longuet-Higgins H. C. (1969) *Nature* **222,** 960

Willshaw D. J. and von der Malsburg C. (1976) *Proc. Roy. Soc. B* **194,** 431

SUBJECT INDEX

0925 hc

ISBN 981-02-0012-9