# 10. Associative Memory: Advanced Learning Strategies

## 10.1 Storing Correlated Patterns

As we discussed in Sects. 3.1 and 4.3, the ability to recall memories correctly breaks down if the number $p$ of stored patterns exceeds a certain limit. When the synaptic connections are determined according to Hebb's rule (3.12), this happens at the storage density $\alpha = p/N = 0.138$. The reason for this behavior was the influence of the other stored patterns as expressed by the fluctuating noise term in (3.13). As we already pointed out at the end of Sect. 3.1, this influence vanishes exactly if the patterns are orthogonal to each other as defined in (3.16). On the other hand, the power of recollection deteriorates even earlier if the stored patterns are strongly correlated. Unfortunately, this happens in many practical examples. Just think of the graphical representation of roman letters, where "E" closely resembles "F" and "C" resembles "G", or of a typical list of names from the telephone book, which are probably highly correlated.

Use the program ASSO (see Chapt. 22) to learn and recall the 26 letters of the alphabet: A–Z. Choose the following parameter values: (26/26/0/1) and (1/0/0/1;2), i.e. sequential updating, temperature and threshold zero, and experiment with the permissible amount of noise. Is any letter stable? Repeat the exercise with the first six letters of the alphabet and study the network's ability to recall the similar letters "E" and "F".

The nature of the problem is not so different from that encountered in the previous section in connection with layered feed-forward networks. The perceptron learning rule provides the perfect learning strategy for simple perceptrons without hidden layers of neurons. However, these devices are not particularly useful in practice, since they fail to solve even some very simple tasks. This is the reason why the perceptron concept fell out of grace for almost twenty years, although multilayered perceptrons with hidden neurons do not suffer from such ailments. But without a practical learning algorithm,

which became first available with error back-propagation, they did not provide a practical alternative.

Similarly, some of the practical difficulties encountered with associative-memory networks are not of a fundamental nature, but rather a consequence of the inadequacy of the elementary form (3.12) of Hebb's learning rule. As we discussed in Sect. 3.1, Hebb's rule is based on the concept of the Hamming measure (3.1) of distance between different patterns. In mathematical terms, this distance measure or *metric* is called the *Euclidean* metric in the space of patterns. More general measures of the distance between different patterns are conceivable and may be more useful if the patterns are correlated. For example, in the case of the letters "E" and "F" a distance measure based solely on the bottom part of the letter would easily discriminate between the patterns. However, as the letters "K" and "R" show, this simple choice does not yet provide a general solution, not even for the alphabet.

### 10.1.1 The Projection Rule

Nonetheless, it turns out that the problem of discriminating between correlated patterns has a remarkably simple solution, which even permits the storage of $p = N$ arbitrarily correlated patterns, as long as they are linearly independent. To see how it works, we form the matrix of scalar products between all pairs of patterns ($\sigma_i^\mu = \pm 1$):

$$Q_{\mu\nu} = \frac{1}{N} \sum_i \sigma_i^\mu \sigma_i^\nu \qquad (1 \le \mu, \nu \le p) \,. \tag{10.1}$$

For linearly independent patterns the matrix $Q_{\mu\nu}$ is invertible, and we can define the following improved synaptic coupling strengths [Ko84, Pe86b]:

$$\tilde{w}_{ij} = \frac{1}{N} \sum_{\mu,\nu} \sigma_i^\mu (Q^{-1})_{\mu\nu} \sigma_j^\nu \,. \tag{10.2}$$

Mathematically, (10.2) corresponds to a projection technique that eliminates the existing correlations between patterns, hence this learning rule is often called the *projection rule*. With this choice the interaction of the stored patterns due to the fluctuating term in (3.13) vanishes exactly, as can be easily seen by computing the post-synaptic potentials in the presence of one of the memorized patterns $\sigma_i^\lambda$:

$$\begin{aligned} \tilde{h}_i = \sum_j \tilde{w}_{ij} \sigma_j^\lambda &= \frac{1}{N} \sum_{\mu,\nu} \sigma_i^\mu (Q^{-1})_{\mu\nu} \sum_j \sigma_j^\nu \sigma_j^\lambda \\ &= \sum_{\mu,\nu} \sigma_i^\mu (Q^{-1})_{\mu\nu} Q_{\nu\lambda} = \sum_\mu \sigma_i^\mu \delta_{\mu\lambda} = \sigma_i^\lambda . \end{aligned} \tag{10.3}$$

We conclude that every stored pattern represents a stable network configuration, independent of correlations among the patterns. Of course, the condition $p \le N$ continues to limit the memory capacity, since at most $N$ linearly independent patterns can be formed from $N$ units of information.

As it stands this statement is not entirely correct because some patterns may be effectively memorized without being explicitly represented in the synaptic couplings. Such a phenomenon is not new to to us; we have seen in Sect. 3.3 (3.23) that linear combinations of stored patterns may also be stable memory states. As Opper has shown [Op88] this occurs for the *iterative learning algorithm* of Krauth and Mézard [Kr87a] (see also Sect. 10.1.3), which permits the storage of up to $2N$ different patterns in an optimal way. Only $N$ of these patterns are stored according to the projection rule, the others are memorized without being explicitly stored. However, for $p > N$ the learning process converges very slowly. (Techniques to accelerate convergence were suggested in [Ab89, An89a].) The optimal storage capacity of a neural network will be discussed in detail in Chapt. 20.

The prescription (10.2) for the coupling strengths is also called the *pseudoinverse solution*. Essentially it performs an inversion of the set of pattern vectors $\sigma_i^\mu$ which can be viewed as a matrix with $N$ columns and $p$ rows. The role of the pseudoinverse is most easily understood if we look at (10.3) at a fixed site, dropping the index $i$. Then we have $\tau^\mu = \sum_j \sigma^\mu{}_j \, w_j$ or in matrix notation $\boldsymbol{\tau} = \underline{\sigma}\, \mathbf{w}$. Here $\tau^\mu$ stands for the output which is to be evoked when the input pattern $\sigma_j^\mu$ is presented to the network. We have to solve a system of $p$ linear equations for the $N$-dimensional weight vector $\mathbf{w}$. Since in general the matrix $\underline{\sigma}$ is not square ($p < N$) we cannot invert it directly. However, one can introduce the pseudoinverse matrix [Ko84]

$$\underline{\sigma}^{\mathrm{pi}} = \underline{\sigma}^T \left( \underline{\sigma}\, \underline{\sigma}^T \right)^{-1} \tag{10.4}$$

so that $\mathbf{w} = \underline{\sigma}^{\mathrm{pi}} \boldsymbol{\tau}$, (cf. (10.2)), solves the problem:

$$\underline{\sigma}\mathbf{w} = \underline{\sigma}\, \underline{\sigma}^{\mathrm{pi}} \boldsymbol{\tau} = \underline{\sigma}\, \underline{\sigma}^T \left( \underline{\sigma}\, \underline{\sigma}^T \right)^{-1} \boldsymbol{\tau} = \boldsymbol{\tau} \, . \tag{10.5}$$

In an autoassociative memory $\tau$ happens to coincide with $\sigma$ but this is not essential, the pseudoinverse solution also works for heteroassociation and for perceptrons without hidden layers.

The quality of pattern recall deteriorates with growing temperature $T$ and memory utilization $\alpha = p/N$ [Ka87].[1] As in the case of Hebb's rule the quality of recollection is described by the parameter $m$ defined in (4.15). $m = 1$ denotes perfect memory recall, whereas $m = 0$ indicates total amnesia. The regions of working and confused memory are shown, together with the value of $m$ at the phase boundary, in Fig. 10.1. The *radius of attraction* $R$ of the stored patterns is shown in Fig. 10.2 as function of storage density $\alpha$ for the models of Kanter and Sompolinsky [Ka87] (curve a) and Personnaz et al. [Pe86b] (curve b). Here the radius of attraction is defined as $R = 1 - m_0$, where $m_0$ is the smallest overlap a pattern $s_i$ can have with a stored pattern $\sigma_i$ to be recognized with certainty by the network. As one sees, the elimination

[1] Note that the diagonal couplings $\tilde{w}_{ii}$ are set to zero in [Ka87]. If the diagonal terms are retained [Pe86b], the critical memory capacity remains $\alpha_c = 1$ at $T = 0$, but in the presence of tiny fluctuations the stored patterns cannot be recalled above $\alpha = 0.5$. (One says that the *radius of attraction* of the stored patterns is zero.)

of the diagonal couplings $w_{ii}$ in the model of [Ka87], curve (a), has a very beneficial effect.

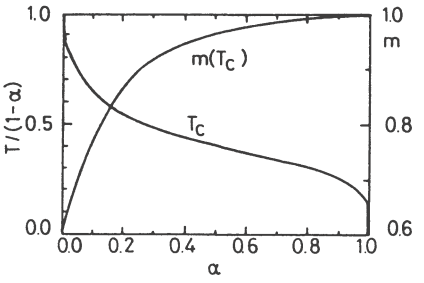



**Fig. 10.1.** Regions of working memory and total confusion, and recall quality $m$ at the phase boundary labeled "$T_c$".

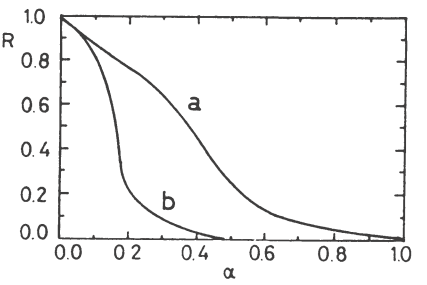



**Fig. 10.2.** Average radius of attraction $R$ as function of storage density in the models [Ka87] (*curve a*) and [Pe86b] (*curve b*).

### 10.1.2 An Iterative Learning Scheme

The practical application of the projection learning rule for large, memory-saturated networks suffers from the need to invert the $(p \times p)$-matrix $Q_{\mu\nu}$, which poses a formidable numerical problem. Fortunately, the matrix inversion need be performed only once, when the patterns are stored into the network. The ingrained memory can then be recalled as often as desired without additional effort. A practical method of implementing the projection rule is based on an iterative scheme, where the "correct" synaptic connections are strengthened in order to stabilize the correlated patterns against each other [Di87]. For the sake of simplicity, we demonstrate this method only for the deterministic network ($T = 0$).

Because of the neuron evolution law $s_i(t+1) = \mathrm{sgn}[h_i(t)]$ any pattern $\sigma_i^\mu$ represents a stable network configuration, if $h_i$ has the same sign as $\sigma_i$, i.e.

$$\sigma_i^\mu h_i = \sum_j w_{ij}\sigma_i^\mu\sigma_j^\mu > 0 \tag{10.6}$$

for every neuron $i$. If the expression (10.6) is only slightly positive, any small perturbation, i.e. $s_i \neq \sigma_i^\mu$ for a few neurons $i$, can change its sign. In order to achieve greater stability of the desired memory patterns, we demand that the expression (10.6) be not only positive but also greater than a certain threshold $\kappa > 0$. For a single pattern, Hebb's rule (3.7) yields $h_i = \sigma_i$. As a consequence, the condition $\sigma_i h_i = 1$ is always satisfied in this case. It appears natural to take the stability threshold at $\kappa = 1$ also in the general case of several stored patterns, and to demand that the synaptic connections be chosen such that

$$\sigma_i^\mu h_i = \sum_j w_{ij} \sigma_i^\mu \sigma_j^\mu = 1 \tag{10.7}$$

for all neurons $i$.

An obvious method of achieving the desired result begins with choosing the synaptic connections initially according to Hebb's rule:

$$w_{ij} = \frac{1}{N} \sum_\mu \sigma_i^\mu \sigma_j^\mu \ . \tag{10.8}$$

In the next step we check, one after the other for all stored patterns, whether the condition (10.7) is fulfilled. If this is not the case, we modify the synapses according to the prescription

$$w_{ij} \to w'_{ij} = w_{ij} + \delta w_{ij} \tag{10.9}$$

with

$$\delta w_{ij} = \frac{1}{N} \left(1 - \sigma_i^\mu h_i\right) \sigma_i^\mu \sigma_j^\mu \ , \tag{10.10}$$

where $\mu$ denotes the pattern just under consideration.[2] With these modified synaptic connections we obtain for the same pattern $\mu$

$$\begin{aligned} \sigma_i^\mu h'_i &= \sigma_i^\mu h_i + \sum_j \delta w_{ij} \sigma_i^\mu \sigma_j^\mu \\ &= \sigma_i^\mu h_i + \frac{1}{N} \sum_j \left(\sigma_i^\mu\right)^2 \left(\sigma_j^\mu\right)^2 \left(1 - \sigma_i^\mu h_i\right) = 1 \ , \end{aligned} \tag{10.11}$$

since $\left(\sigma_i^\mu\right)^2 = 1$. Thus, after updating all synapses, the threshold stability condition (10.7) is satisfied for the considered pattern. When we proceed to the next pattern $(\mu + 1)$, the synaptic couplings will be modified again, so that (10.7) becomes valid for the pattern now under consideration. However, (10.7) may cease to be satisfied for the previous pattern $\mu$. After a full cycle over all stored patterns the condition is therefore only fulfilled with certainty for the last pattern, $\mu = p$, but not necessarily for all other patterns. The crucial question is whether this updating process converges, or whether it may continue indefinitely without reaching a stationary state, in which the threshold condition is satisfied by all patterns.[3]

In order to study this question, it is useful to introduce some abbreviations. We assume here that the synapses are adjusted sequentially, i.e. after inspection of the performance of the network for each single pattern. We define the deviation from the threshold in the $\ell$th updating cycle, and the sum of all deviations encountered up to that point, as

[2] In principle, we can do without the initial Hebbian choice of synapses. If we start with a completely disconnected network ($w_{ij} = 0$), or *tabula rasa*, the first application of the modification law (10.9) results in synaptic connections with precisely the values assigned by Hebb's rule!

[3] The proof of convergence follows closely that of the perceptron convergence theorem given, e.g., in [Bl62a, Mi69].

$$\delta x_i^\mu(\ell) = 1 - \sigma_i^\mu h_i, \qquad x_i^\mu = \sum_{\ell'=1}^{\ell} \delta x_i^\mu(\ell') \,. \tag{10.12}$$

The synaptic modifications according to (10.10) for the $\nu$th pattern in the $\ell$th cycle can then be written in the form

$$\delta w_{ij} \equiv \frac{1}{N}\left(1 - \sum_k w_{ik}\sigma_i^\nu \sigma_k^\nu\right)\sigma_i^\nu \sigma_j^\nu = \frac{1}{N}\delta x_i^\nu(\ell)\sigma_i^\nu \sigma_j^\nu . \tag{10.13}$$

Thus, after the completion of the $\ell$th updating cycle the synaptic connections can be expressed as

$$w_{ij} = \frac{1}{N}\sum_\nu x_i^\nu(\ell)\sigma_i^\nu \sigma_j^\nu \,. \tag{10.14}$$

Consider now what happens in the $(\ell+1)$th iteration cycle. If we have just reached the pattern $\mu$, all previous patterns have contributed $(\ell+1)$ times to the synaptic modification process, whereas all others (including pattern $\mu$) have made only $\ell$ contributions. The modification for the $\mu$th pattern is therefore given by

$$\begin{aligned} \delta x_i^\mu &= x_i^\mu(\ell+1) - x_i^\mu(\ell) = 1 - \sum_k w_{ik}\sigma_i^\mu \sigma_k^\mu \\ &= 1 - \frac{1}{N}\sum_k \left[\sum_{\nu<\mu} x_i^\nu(\ell+1)\sigma_i^\nu \sigma_k^\nu \sigma_i^\mu \sigma_k^\mu + \sum_{\nu\geq\mu} x_i^\nu(\ell)\sigma_i^\nu \sigma_k^\nu \sigma_i^\mu \sigma_k^\mu\right]. \end{aligned} \tag{10.15}$$

We now introduce the $N$ matrices of dimension $(p \times p)$

$$B_i^{\mu\nu} = \frac{1}{N}\sum_k \sigma_i^\nu \sigma_k^\nu \sigma_i^\mu \sigma_k^\mu = \sigma_i^\nu \sigma_i^\mu Q_{\mu\nu}, \qquad (i = 1, \ldots, N), \tag{10.16}$$

where $Q_{\mu\nu}$ is the symmetric overlap matrix defined in (10.1). This allows us to put (10.15) into the simple form

$$x_i^\mu(\ell+1) - x_i^\mu(\ell) = 1 - \sum_{\nu<\mu} B_i^{\mu\nu} x_i^\nu(\ell+1) - \sum_{\nu\geq\mu} B_i^{\mu\nu} x_i^\nu(\ell) \,. \tag{10.17}$$

Assuming that the iteration procedure converges, i.e.

$$\lim_{\ell\to\infty} x_i^\mu(\ell) = y_i^\mu \,, \tag{10.18}$$

the limiting values must satisfy

$$\sum_\nu B_i^{\mu\nu} y_i^\nu = 1 \tag{10.19}$$

for all values of $i$ and $\mu$. This is a linear system of $Np$ equations for the quantities $y_i^\nu$. The iteration procedure (10.16) is just the well-known Gauss–Seidel method for the iterative solution of a system of linear equations, here the equations (10.19). It can be shown that this method always converges if the matrix $B_i^{\mu\nu}$ has only positive eigenvalues, i.e. if $\sum_{\mu,\nu} B_i^{\mu\nu} z^\mu z^\nu$ is a positive semidefinite quadratic form [St80]. This condition is certainly satisfied in our case, since on account of the definition (10.16):

$$\sum_{\mu,\nu} B_i^{\mu\nu} z^\mu z^\nu = \frac{1}{N} \sum_k \Big( \sum_\nu \sigma_i^\nu \sigma_k^\nu z^\nu \Big)^2 \geq 0 \,. \tag{10.20}$$

We conclude that the iteration process is guaranteed to converge, yielding the synaptic connections

$$w_{ij} \to \overline{w}_{ij} = \frac{1}{N} \sum_\nu y_i^\nu \sigma_i^\nu \sigma_j^\nu \,. \tag{10.21}$$

Owing to the relation (10.16) between the matrices $B_i^{\mu\nu}$ and $Q_{\mu\nu}$ we can write the equation (10.19) for $y_i^\nu$ also in the form

$$\sum_\nu Q_{\mu\nu} \sigma_i^\nu y_i^\nu = \sigma_i^\mu \,, \tag{10.22}$$

where we have multiplied by $\sigma_i^\mu$ and made use of the property $(\sigma_i^\mu)^2 = 1$. Multiplying with the inverse of the matrix $Q_{\mu\nu}$ and utilizing the same relation we find

$$y_i^\nu = \sigma_i^\nu \sum_\mu (Q^{-1})_{\nu\mu} \sigma_i^\mu \,. \tag{10.23}$$

Upon inserting this into (10.21), which describes the synaptic strengths at the end of the iteration process, we obtain the result

$$\overline{w}_{ij} = \frac{1}{N} \sum_{\mu,\nu} (Q^{-1})_{\mu\nu} \sigma_i^\mu \sigma_j^\nu \equiv \tilde{w}_{ij} \,. \tag{10.24}$$

These are precisely the synaptic connections (10.2), $\tilde{w}_{ij}$, of the projection rule discussed at the beginning of this section, which solve the problem of storing correlated patterns.

### 10.1.3 Repeated Hebbian Learning

For most practical purposes it is not necessary to use precisely the optimal synaptic couplings $\tilde{w}_{ij}$, or, in other words, it is not essential to render the left-hand side of (10.7) exactly equal to one. We recall that the starting point of our considerations was the desire to make the expression $\sigma_i^\mu h_i$ significantly greater than the critical-stability threshold zero. This condition is also satisfied if we modify the synaptic connection in such a way that the left-hand side of (10.7) is *greater than* or *equal to* a given threshold $\kappa$, which may or may not be taken equal to 1. This condition has the important advantage that the iteration process is guaranteed to come to an end after a finite number of steps, and it yields a maximal memory capacity $\alpha_c = 2$.

The procedure then works exactly as described above, except that we strengthen all synapses of a "subcritical" neuron by a fixed amount $1/N$, i.e. we replace the expression (10.10) by [Di87, Kr87a, Ga88a, Fo88b]

$$\delta w_{ij} = \frac{1}{N} \sigma_i^\mu \sigma_j^\mu \, (1 - \delta_{ij}) \, \theta(\kappa - \gamma_i^\mu) \,, \tag{10.25}$$

with the step function $\theta(x)$ and the normalized stability measures[4]

$$\gamma_i^\mu = \sigma_i^\mu h_i / \|w_i\|, \qquad \|w_i\| = \left( \sum_j^{j \neq i} w_{ij}^2 \right)^{1/2} . \tag{10.26}$$

Here we have explicitly dropped the synaptic self-couplings $w_{ii}$, which leads to a better performance of the memory, as discussed in the previous subsection. The same calculation as in (10.11) then yields

$$\gamma_i^{\mu\prime} = \gamma_i^\mu + \theta(\kappa - \gamma_i^\mu) . \tag{10.27}$$

When this expression is larger than $\kappa$, the iteration has converged; otherwise the synaptic reinforcement must be repeated.

In a sense this procedure can be understood as repeated learning according to Hebb's rule, where the synapses are increased by the amount $\frac{1}{N}\sigma_i^\mu \sigma_j^\mu$ as often as necessary to obtain the required stability for all stored patterns. This procedure reminds one of the experience of learning new words of a foreign language, where it is usually necessary to repeat those words several times until they have entered the long-term memory. As everyone knows, this method works for sure – if only after an annoyingly large number of repetitions!

It is therefore important to optimize the learning rate as much as possible. For this aim Abbott and Kepler [Ab89] have modified (10.25) by introducing a new function $f(\gamma)$ that modulates the magnitude of synaptic change according to the remaining deviation from the desired stability goal:

$$\delta w_{ij} = \frac{1}{N} \sigma_i^\mu \sigma_j^\mu \, (1 - \delta_{ij}) \, f\left(\gamma_i^\mu\right) \|w_i\| \theta\left(\kappa - \gamma_i^\mu\right) . \tag{10.28}$$

Two choices of this function were considered, namely the quasilinear function

$$f_{\mathrm{L}}(\gamma) = (\kappa + \delta - \gamma)\theta(\kappa + \delta - \gamma) - 2\gamma\theta(-\kappa - \delta - \gamma) \tag{10.29}$$

and the nonlinear function

$$f_{\mathrm{NL}}(\gamma) = (\kappa + \delta - \gamma) + \sqrt{(\kappa + \delta - \gamma)^2 - \delta^2} . \tag{10.30}$$

Here $\delta \ll 1$ is a parameter that controls the speed of learning. For the quasilinear function (10.29) the algorithm converges after less than $2N/\delta^2$ iterations.

The rate of convergence for a typical simulation with the parameters $\kappa = 0.43$, $\delta = 0.01$, and $N = 100$ is shown in Fig. 10.3. The storage density was $\alpha = 0.75$, i.e. 75 patterns were to be stored by the network. Curve (a) refers to the standard algorithm (10.25), while curves (b) and (c) refer to the optimized algorithm (10.28) with the functions $f_{\mathrm{L}}$ and $f_{\mathrm{NL}}$, respectively. The advantage of the modified algorithm is obvious. An analytic expression for

[4] The various references given above deviate slightly in their definition of the stability measure. The interested reader is urged to consult the original literature for details. We also refer to Chapt. 22 where some additional information on the learning rules can be found.

the rate of convergence of the standard algorithm (10.25) has been derived by Opper [Op88], which shows that the convergence slows down dramatically when the critical memory density is approached.

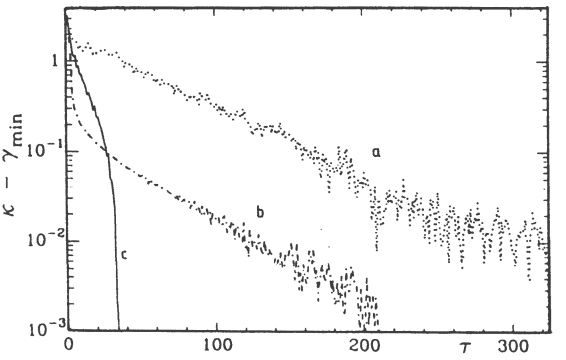



**Fig. 10.3.** Rate of convergence for the iterative learning rules (10.25) (*curve a*) and (10.28), (*curves b and c*). $\tau$ counts the number of iterations (from [Ab89]).

How large should the stability threshold $\kappa$ be chosen? If $\kappa$ is taken too large, no solution of the stability condition $\gamma_i^\mu > \kappa$ may exist. Then the algorithm will not converge. If $\kappa$ is taken too small, the algorithm (10.25) converges rapidly and the stored patterns are stable, but the basins of attraction are small and the neural network will not necessarily recognize a slightly perturbed pattern. A certain amount of experience is required to find the optimal set of learning parameters. The dependence of the average radius of attraction of the stored patterns on the choice of $\kappa$ was studied by Kepler and Abbott [Ke88], who found that it drops to $R = 0.1$ at $\kappa = 1$ for a saturated network.

Use the program ASSO (see Chapt. 22) to learn all 26 letters of the alphabet with any of the improved learning schemes 2–5. Compare their speed of convergence and the stability of the learned patterns against noise and thermal fluctuations.

Repeat the exercise of Sect. 3.4 with the program ASSCOUNT (cf. Chapt. 23) using the Diederich–Opper learning protocol on up to 10 numbers. Experiment with the parameters governing time delay.

## 10.2 Special Learning Rules

### 10.2.1 Forgetting Improves the Memory!

As we discussed in Sect. 3.3, the standard learning rule (Hebb's rule) leads to the emergence of undesirable local minima in the "energy" functional $E[s]$. In practice, this means that the evolution of the network can be caught in spurious, locally stable configurations, such as those in (3.23). Large networks usually contain a vast number of such spuriously stable states, many of which are not even linear combinations of the desired stability points. The learning rule for correlated patterns discussed in Sect. 10.1 does not guard against this problem. Thermal fluctuations do help to destabilize the spurious configurations, but at the expense of storage capacity. Moreover, the disappearance of all the spurious states at some finite $T$ is not ensured.

A much better strategy is to eliminate the undesired stable configurations by appropriate modifications of the synaptic connections. Hopfield et al. [Ho83] have proposed to make use of the fact that the spurious minima of the energy functional $E[s]$ are usually much shallower than the minima that correspond to the learned patterns. Borrowing ideas developed in the study of human dream sleep, and discussed in Sect. 2.3, they suggested tracking these states by starting the network in some randomly chosen initial configuration and running it until it ends up in a stable equilibrium state $s_i^\infty$. This may be one of the regular learned patterns, or one of the many spurious states. Whatever the resulting state is, the synapses are partially *weakened* according to Hebb's rule:

$$w_{ij} \to w_{ij} - \frac{\lambda}{N} s_i^\infty s_j^\infty \ , \tag{10.31}$$

where $\lambda \ll 1$ is chosen. This procedure of unlearning has two favorable effects. Most spurious equilibrium states of the network are "forgotten", since they are already destabilized by small changes in the synaptic connections $w_{ij}$. Moreover, the different regions of stability of the stored patterns become more homogeneous in size, since those with a larger range of stability occur more often as final configurations and are therefore weakened more than others.

The effect of this intentional forgetting is especially apparent in the sizes of the basins of attraction. This term denotes the set of all states, from which the network dynamics leads to a particular pattern. The change of the size of the basin of attraction of a given stored pattern, as the total memory load is increased, is illustrated in Fig. 10.4. The two axes labeled $H_k$ and $H_{N-k}$ in these figures represent a crude measure of the distance of an initial trial state $s_i$ from the considered memory state $\sigma_i^\mu$. They denote the partial Hamming distances between the trial state and the memory state, evaluated for the first $k$ and the last $(N-k)$ of all $N = 200$ neurons, respectively.

$$H_k = \frac{1}{4}\sum_{i=1}^{k}(s_i - \sigma_i^\mu)^2\,, \qquad H_{N-k} = \frac{1}{4}\sum_{i=k}^{N-k}(s_i - \sigma_i^\mu)^2\,. \tag{10.32}$$

In this specific case $k = N/2$ was taken, i.e. the axes represent the Hamming distance for the first and the last half of the neurons of the network. If the trial state $s_i$ developed into the stored pattern $\mu$, a black dot was plotted. For a single memory state (Fig. 10.4a), half of the trial states are found to evolve into the stored pattern $\sigma_i$, the other half ends up in the complementary pattern $(-\sigma_i)$. The basin of attraction thus represents a black triangle. For more memory states this region shrinks rapidly and takes on a highly ragged shape in the vicinity of $\alpha_c$, as shown in Fig. 10.4b,c for 28 and 32 uncorrelated memory states, respectively. Figure 10.4d shows the result of applying the forgetting algorithm (10.31) 1000 times to the network loaded with 32 patterns [Ke87]. (The unlearning strength was $\lambda = 0.01$.) The basin of attraction grows strongly (by a factor of ten or more) and also takes on a more regular shape. The probability of retrieving the stored patterns is much improved.

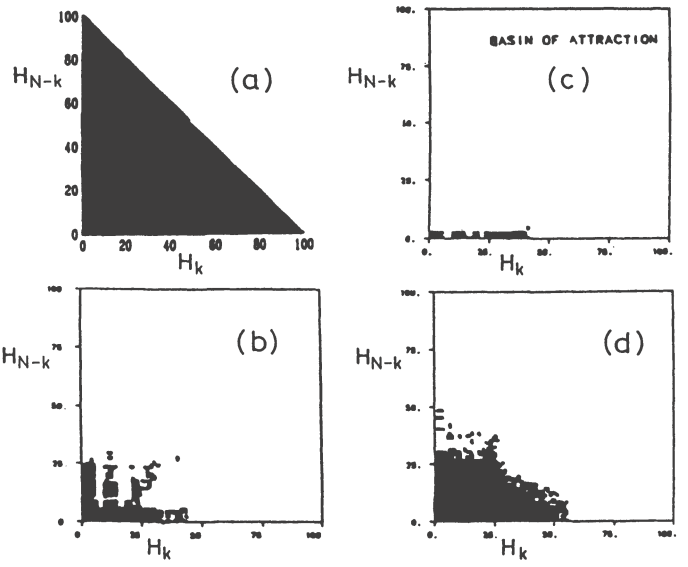



**Fig. 10.4.** Basins of attraction in a Hopfield network with 200 neurons for (**a**) 1, (**b**) 28, (**c**) 32 memory states. After deliberate forgetting the basin expands strongly (**d**). (From [Ke87]).

In a somewhat modified version of this method [Pö87] the network is allowed to develop from the stored patterns, deteriorated by random noise. One then not only weakens the synaptic connections by unlearning the final state $s_i^\infty$, but also simultaneously relearns the correct starting pattern $\nu$:

$$w_{ij} \to w_{ij} - \frac{\lambda}{N}\left(s_i^\infty s_j^\infty - \sigma_i^\nu \sigma_j^\nu\right) . \tag{10.33}$$

If the pattern was recalled without fault, the synapses remain unchanged according to this prescription. With this method the storage capacity can be increased to $\alpha = 1$, and the storage of strongly correlated patterns becomes possible.

Unfortunately, these methods do not eliminate the spurious stable states corresponding to linear combinations of stored patterns, such as (3.23). The total synaptic modification for all eight of these states taken together vanishes exactly, since the sum of the changes (10.31) adds to zero. However, a slightly different procedure works successfully [Ki87], where forgetting is controlled by the rule

$$w_{ij} \to w_{ij} - \frac{\eta}{N}\left(\sigma_i^\mu \sigma_i^\nu \sigma_i^\lambda\right)\left(\sigma_j^\mu \sigma_j^\nu \sigma_j^\lambda\right) . \tag{10.34}$$

Here $\mu$, $\nu$, and $\lambda$ denote any triple of stored patterns. For $\eta > 1/3$ one finds that all eight spurious states (3.23) already become unstable at $T = 0$. At a finite value of the temperature parameter a smaller value of $\lambda$ suffices for destabilization, and above $T = 0.46$ these spurious configurations become unstable because of the action of thermal fluctuations alone.

### 10.2.2 Nonlinear Learning Rules

An essential disadvantage of Hebb's rule (3.12) for $p$ patterns

$$w_{ij} = \frac{1}{N}\sum_{\mu=1}^{p} \sigma_i^\mu \sigma_j^\mu \tag{10.35}$$

for many applications is that the synaptic strengths can vary over a wide range $-p/N \leq w_{ij} \leq +p/N$ . This is particularly disturbing in hardware implementations of neural networks, since it requires electronic switching elements with a wide dynamical range. It is therefore natural to ask whether the range of allowed values of the $w_{ij}$ can be limited with impunity. The extreme case would be to distinguish only between excitatory ($w_{ij} > 0$) and inhibitory ($w_{ij} < 0$) synapses, which are assigned the same absolute strength, but different signs:

$$w_{ij} = \frac{\sqrt{p}}{N}\mathrm{sgn}\left(\sum_{\mu=1}^{p} \frac{1}{\sqrt{p}}\sigma_i^\mu \sigma_j^\mu\right) = \pm\frac{\sqrt{p}}{N} . \tag{10.36}$$

This procedure, called *clipping* of synapses, is a special case of the nonlinear Hebb rule [He86, He87c, So86b], (see also Sect. 19.3)

$$w_{ij} = \frac{\sqrt{p}}{N}\Phi\left(\frac{1}{\sqrt{p}}\sum_{\mu=1}^{p} \sigma_i^\mu \sigma_j^\mu\right) , \tag{10.37}$$

where $\Phi(x)$ is an arbitrary, monotonously increasing function. The choice $\Phi(x) = x$ leads back to the standard (linear) Hebb rule (3.12), while $\Phi(x) = \mathrm{sgn}(x)$ describes clipped synapses.

Clipping of synapses has a surprisingly small influence on the storage capacity of a network and on its ability to recall stored patterns. Compared to Hebb's linear rule the learning rule (10.36) acts as additional noise, causing a reduction of the critical storage density $\alpha_c = p_{\max}/N$ at $T = 0$ from ${\alpha_c}^{\mathrm{Hebb}} = 0.138$ to $\alpha_c = 0.102$. It can be shown that in general the memory capacity is always less than in the linear Hebb case, ${\alpha_c}^{\Phi} \leq {\alpha_c}^{\mathrm{Hebb}}$, for arbitrary synapses [He87c, He88a]. At low memory density, i.e. for small values of $\alpha$, the error rate $(1 - m)/2$ in pattern recall is insignificantly larger for clipped synapses than for Hebbian ones. These deteriorations are compensated by the important simplifications resulting for the storage of the synaptic connections $w_{ij}$ in hardware realizations, as well as in software simulations of the neural network on conventional digital computers.

Of particular interest are bounded synaptic strength functions $|\Phi(x)| \leq \Phi_0$. Here it is useful to modify the learning rule (10.37), in order to allow for the addition of more and more patterns to the memory: if the synapses after storing $(\mu - 1)$ patterns are denoted by $w_{ij}^{(\mu-1)}$, those obtained after adding the next pattern to the memory are defined as

$$w_{ij}^{(\mu)} = \Phi\Big(\epsilon\sigma_i^\mu\sigma_j^\mu + w_{ij}^{(\mu-1)}\Big) \,. \tag{10.38}$$

Whereas in the case of the linear Hebb rule the continued addition of more memory states eventually leads to the complete breakdown of the ability of the network to retrieve any stored pattern, memory degradation proceeds in a much gentler way for the nonlinear learning law (10.38) with bounded synaptic strengths. As more and more patterns are stored, the network approaches the limit of its storage capacity, $\alpha_c$. However, instead of entering a state of total confusion, the network then experiences a gradual "blurring" of the older memory states, which are slowly replaced by the freshly learned patterns.

Because the whole memory can eventually be viewed as a sequence of clearly retrievable fresh patterns superimposed on increasingly deteriorated older patterns, such a memory structure is often called a *palimpsest*.[5] Memories of this type can serve as efficient *short-term memories*, because they permit the network to function as an information storage device continuously without encountering its capacity limit. Besides overwhelming physiological evidence that short-term-memory structures exist in the brain, such memories have important applications in electronic information processing as so-called *cache* memories.

[5] The term *palimpsest* derives from a practice used in the Middle Ages, when parchment was so precious that it was written upon several times, the earlier writing having been wholly or partially erased to make room for the next.

The nonlinear learning rule (10.38) was studied by Parisi [Pa86b] for the bounded function

$$\Phi(x) = \begin{cases} x & \text{for } |x| \leq 1 \\ \text{sgn}(x) & \text{for } |x| > 1 \end{cases} \tag{10.39}$$

shown in Fig. 10.5. As expected, the recall quality $m$ of a stored state deteriorates after the addition of many other patterns to the memory, as depicted in Fig. 10.6. Although the total memory capacity remains bounded at any one time (Parisi found $\alpha_{\max} = p_{\max}/N \approx 0.04$), the network never loses its ability to learn new patterns. Similar models were studied by Nadal et al. [Na86] and various other authors [Ge87, He88c, Ge89]. A general discussion of their properties can be found in [Mo88].

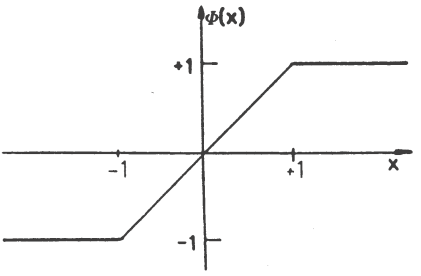



**Fig. 10.5.** Bounded synaptic-strength function used for palimpsest memory networks.

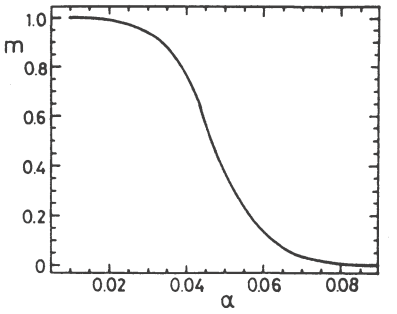
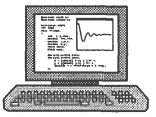



**Fig. 10.6.** The recall quality $m$ of an older memory state fades after the addition of $p = \alpha N$ new states (from [Pa86b]).



Use the program ASSO (see Chapt. 22) to learn at least ten letters of the alphabet with any of the improved learning schemes 2–5, using standard values for the other parameters. Then choose option "m" in the search menu to limit the synaptic strength, or to allow only for binary synapses.

Use the program ASSO to learn at least ten letters of the alphabet with any of the improved learning schemes 2–5, using standard values for the other parameters, except selecting only positive (excitatory) or only negative synapses on the first screen.

### 10.2.3 Dilution of Synapses

The assumption that all neurons are interconnected is not very realistic, especially if one is interested in modeling biological neural nets. One may ask whether a neural network can still act as an efficient associative memory if a large fraction of the synaptic connections are severed or *diluted.* The synaptic connectivity is usually diluted by eliminating connections at random, keeping only a fraction $d < 1$ of all synapses [So86b]. If this is done while the symmetry of the synaptic matrix $w_{ij}$ is preserved, the memory capacity $\alpha_c$ of a network trained with Hebb's rule drops almost linearly with $d$, as illustrated in Fig. 10.7, where $\alpha_c$ and the critical recall quality $m_c = m(\alpha_c)$ are shown as functions of the fraction of destroyed synaptic connections.[6] This result clearly exhibits the *error resistivity* of neural networks. Even after elimination of a large fraction of synapses the network continues to operate quite reliably. Another approach is to dilute the synapses asymmetrically, i.e. to set $w_{ij} = 0$ but not necessarily also require $w_{ji} = 0$. This case can be treated analytically in the limit $d \to 0$ [De87c, Cr86]. One finds that the memory capacity per remaining synapse is about four times as large, $\alpha_c = 2d/\pi \approx 0.64d$, as for a fully connected network.

Virasoro has pointed out that the random destruction of synapses can lead to interesting effects when the stored patterns have a hierarchical similarity structure [Vi88]. By this one means that the patterns fall into several distinct classes or categories, the patterns belonging to a common class being strongly correlated [Pa86a, Do86]. A set of patterns with this property can be generated in the following way. First choose $Q$ uncorrelated class patterns $\xi_i^\alpha, (i = 1, \ldots, N; \alpha = 1, \ldots, Q)$, where the $\xi_i^\alpha = \pm 1$ with equal probability. For each category $\alpha$ one now generates $p_\alpha$ correlated patterns $\sigma_i^{\alpha\mu}, (i = 1, \ldots, N; \mu = 1, \ldots, p_\alpha)$, taking the value $\sigma_i^{\alpha\mu} = \pm\xi_i^\alpha$ with probability $(1 \pm m)/2$. In the limit $m \to 1$ the patterns within the same category become more and more similar.

When the memory capabilities of the network deteriorate, e.g. because of overloading, the presence of thermal noise, or synaptic dilution, the network may reach a stage at which an individual pattern $\sigma_i^{\alpha\mu}$ can no longer be retrieved, but the recall of the corresponding class $\xi_i^\alpha$ is still possible.[7] Such a behavior can be of great interest, because the network can then perform the task of categorization, i.e. identify the class to which a given pattern that is

[6] This implies that the storage efficiency of the network does not really decrease, since fewer synapses are required in proportion. One has to keep in mind here that the complexity of the neural system is not described by the number $N$ of neurons but by the total number of synaptic connections, $\frac{1}{2}dN(N-1)$, so that the true memory efficiency is given by $\alpha_c/d$.

[7] The visual inability to recognize the difference between similar objects, e.g. human faces, is called *prosopagnosia* in clinical psychology. It is distinguished from the syndrome of *agnosia*, where the whole act of visual recognition is impaired, and which has been vividly described by Sacks [Sa87].