

be perfect. Effectively, a pattern can be recalled or recognized because of associations formed between its parts. This of course requires distributed representations.

Next we introduce a more precise and detailed description of the above, and describe the properties of these networks. Ways to analyse formally the operation of these networks are introduced in Appendix A4.

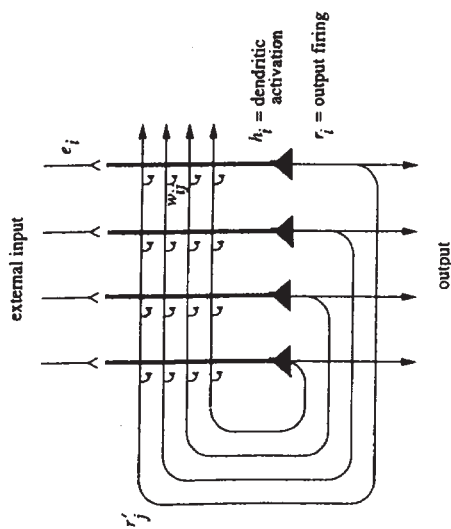


Fig. 3.1 The architecture of an autoassociative neural network.

3.1.1 Learning

The firing of every output neuron i is forced to a value r_i determined by the external input e_i . Then a Hebb-like associative local learning rule is applied to the recurrent synapses in the network:

$$\delta w_{ij} = k r_i r_j \quad (\text{Hebb rule}) \quad (3.1)$$

It is notable that in a fully connected network, this will result in a symmetric matrix of synaptic weights, that is the strength of the connection from neuron 1 to neuron 2 will be the same as the strength of the connection from neuron 2 to neuron 1 (both implemented via recurrent collateral synapses).

It is a factor which is sometimes overlooked that there must be a mechanism for ensuring that during learning r_i does approximate e_i , and must not be influenced much by activity in the recurrent collateral connections, otherwise the new external pattern e will not be stored in the network, but instead something will be which is influenced by the previously stored memories. It is thought that in some parts of the brain, such as the hippocampus, there are processes which help the external connections to dominate the firing during learning (see Chapter 6 and Rolls, 1989b-e; Treves and Rolls, 1992).

3 Autoassociation memory

Autoassociative memories, or attractor neural networks, store memories, each one of which is represented by a pattern of neural activity. They can then recall the appropriate memory from the network when provided with a fragment of one of the memories. This is called completion. Many different memories can be stored in the network and retrieved correctly. The network can learn each memory in one trial. Because of its 'one-shot' rapid learning, and ability to complete, this type of network is well suited for episodic memory storage, in which each past episode must be stored and recalled later from a fragment, and kept separate from other episodic memories. An autoassociation memory can also be used as a short term memory, in which iterative processing round the recurrent collateral connection loop keeps a representation active until another input cue is received. In this short term memory role, it appears to be used in the temporal visual cortical areas with their connections to the ventrolateral prefrontal cortex for the short term memory of visual stimuli (in delayed match to sample tasks, see Chapters 6 and 8); and in the dorsolateral prefrontal cortex for short term memory of spatial responses (see Chapter 10). A feature of this type of memory is that it is content addressable; that is, the information in the memory can be accessed if just the contents of the memory (or a part of the contents of the memory) are used. This is in contrast to a conventional computer, in which the *address* of what is to be accessed must be supplied, and used to access the contents of the memory. Content addressability is an important simplifying feature of this type of memory, which makes it suitable for use in biological systems. The issue of content addressability will be amplified below.

3.1 Architecture and operation

The prototypical architecture of an autoassociation memory is shown in Fig. 3.1. The external input e_i is applied to each neuron i by unmodifiable synapses. This produces firing r_i of each neuron, or a vector of firing on the output neurons r . Each output neuron i is connected by a recurrent collateral connection to the other neurons in the network, via modifiable connection weights w_{ij} . This architecture effectively enables the output firing vector r to be associated during learning with itself. Later on, during recall, presentation of part of the external input will force some of the output neurons to fire, but through the recurrent collateral axons and the modified synapses, other neurons in r can be brought into activity. This process can be repeated a number of times, and recall of a complete pattern may

3.1.2 Recall

During recall the external input e_j is applied, and produces output firing, operating through the non-linear activation function described below. The firing is fed back by the recurrent collateral axons shown in Fig. 3.1 to produce activation of each output neuron through the modified synapses on each output neuron. The internal activation h_i produced by the recurrent collateral effect on the i th neuron is in the standard way the sum of the activations produced in proportion to the firing rate of each axon r_j operating through each modified synapse w_{ij} , that is

$$h_i = \sum_j r_j w_{ij} \quad (3.2)$$

where \sum_j indicates that the sum is over the C input axons to each neuron, indexed by j . The output firing r_i is a function of the activation produced by the recurrent collateral effect (internal recall) and by the external input (e_i):

$$r_i = f(h_i + e_i) \quad (3.3)$$

The activation function should be non-linear, and may be for example binary threshold, linear threshold, sigmoid, etc. (see Fig. 1.3). A non-linear activation function can minimize interference between the pattern being recalled and other patterns stored in the network, and can also be used to ensure that what is a positive feedback system remains stable. The network can be allowed to repeat this recurrent collateral loop a number of times. Each time the loop operates, the output firing becomes more like the originally stored pattern, and this progressive recall is usually complete within 5–15 iterations.

3.2 Introduction to the analysis of the operation of autoassociation networks

With complete connectivity in the synaptic matrix, and the use of a Hebb rule, the matrix of synaptic weights formed during learning is symmetric. The learning algorithm is fast, 'one shot', in that a single presentation of an input pattern is all that is needed to store that pattern.

During recall, a part of one of the originally learned stimuli can be presented as an external input. The resulting firing is allowed to iterate repeatedly round the recurrent collateral system, gradually on each iteration recalling more and more of the originally learned pattern. Completion thus occurs. If a pattern is presented during recall that is similar but not identical to any of the previously learned patterns, then the network settles into a stable recall state in which the firing corresponds to that of the previously learned pattern. The network can thus generalize in its recall to the most similar previously learned pattern. The activation function of the neurons should be non-linear, since a purely linear system would not produce any categorization of the input patterns it receives, and therefore would not be able to effect anything more than a trivial (i.e. linear) form of completion and generalization.

Recall can be thought of in the following way, relating it to what occurs in pattern associators. The external input e is applied, produces firing r , which is applied as a recall cue on the recurrent collaterals as r' . The activity on the recurrent collaterals is then multiplied

with the synaptic weight vector stored during learning on each neuron to produce the new activation h_i which reflects the similarity between r' and one of the stored patterns. Partial recall has thus occurred as a result of the recurrent collateral effect. The activations h_i after thresholding (which helps to remove interference from other memories stored in the network, or noise in the recall cue) result in firing r_i , or a vector of all neurons r , which is already more like one of the stored patterns than, at the first iteration, the firing resulting from the recall cue alone, $r = f(e)$. This process is repeated a number of times to produce progressive recall of one of the stored patterns.

Autoassociation networks operate by effectively storing associations between the elements of a pattern. Each element of the pattern vector to be stored is simply the firing of a neuron. What is stored in an autoassociation memory is a set of pattern vectors. The network operates to recall one of the patterns from a fragment of it. Thus, although this network implements recall or recognition of a pattern, it does so by an association learning mechanism, in which associations between the different parts of each pattern are learned. These memories have sometimes been called autocorrelation memories, because they learn correlations between the units of the network, in the sense that each pattern learned is defined by a set of simultaneously active neurons; this is provided for by the Hebb-like learning rule.

The system formally resembles spin glass systems of magnets analysed quantitatively in statistical mechanics. This has led to the analysis of (recurrent) autoassociative networks as dynamical systems made up of many interacting elements, in which the interactions are such as to produce a large variety of basins of attraction of the dynamics. Each basin of attraction corresponds to one of the originally learned patterns, and once the network is within a basin it keeps iterating until a recall state is reached which is the learned pattern itself or a pattern closely similar to it (interference effects may prevent an exact identity between the recall state and a learned pattern). This type of system is contrasted with other, simpler, systems of magnets (e.g. ferromagnets), in which the interactions are such as to produce only a limited number of related basins, since the magnets tend to be, for example, all aligned with each other. The states reached within each basin of attraction are called attractor states, and the analogy between autoassociator neural networks and physical systems with multiple attractors was drawn by Hopfield (1982) in a very influential paper. He was able to show that the recall state can be thought of as the local minimum in an energy landscape, where the energy would be defined, in our notation, as

$$E = -1/2 \sum_{ij} w_{ij} (r_i - \langle r \rangle) (r_j - \langle r \rangle) \quad (3.4)$$

This equation can be understood in the following way. If two neurons are both firing above their mean rate (denoted by $\langle r \rangle$), and are connected by a weight with a positive value, then the firing of these two neurons is consistent with each other, and they mutually support each other, so that they contribute to the system's tendency to remain stable. If across the whole network such mutual support is generally provided, then no further change will take place, and the system will indeed remain stable. If, on the other hand, either of our pair of neurons were not firing, or if the connecting weight had a negative value, the neurons would not support each other, and indeed the tendency would be for the neurons to try and alter ('flip' in the case of binary units) the state of the other. This would be repeated across the whole

network until a situation in which most mutual support, and least 'frustration', was reached. What makes it possible to define an energy function and to make these simple considerations is that the matrix is symmetric (see Hopfield, 1982; Hertz, Krogh and Palmer, 1991; Amit, 1989).

Physicists have generally analysed a system in which the input pattern is presented and then immediately removed, so that the network then 'falls' without further assistance (a what is referred to as the *unclamped* condition) towards the minimum of its basin of attraction. A more biologically realistic system is one in which the external input is left contributing to the recall during the fall into the recall state. In this *clamped* condition recall is usually faster, and more reliable, so that more memories may be usefully recalled from the network. The approach using methods developed in theoretical physics has led to rapid advances in the understanding of autoassociative networks, and its basic elements are described in Appendix A4.

3.3 Properties

The internal recall in autoassociation networks involves multiplication of the firing vector of neuronal activity by the vector of synaptic weights on each neuron. This inner product vector multiplication allows the similarity of the firing vector to previously stored firing vectors to be provided by the output (as effectively a correlation), if the patterns learned are distributed. As a result of this type of correlation computation performed if the patterns are distributed many important properties of these networks arise, including pattern completion (because part of a pattern is correlated with the whole pattern), and graceful degradation (because damaged synaptic weight vector is still correlated with the original synaptic weight vector). Some of these properties are described next.

3.3.1 Completion

Perhaps the most important and useful property of these memories is that they complete an incomplete input vector, allowing recall of a whole memory from a small fraction of it. If memory recalled in response to a fragment is that stored in the memory that is closest to pattern similarity (as measured by the dot product, or correlation). Because the recall is iterative and progressive, the recall can be perfect.

This property and the associative property of pattern associator neural networks are very similar to the properties of human memory. This property may be used when we recall a part of a recent memory of a past episode from a part of that episode. The way in which this could be implemented in the hippocampus is described in Chapter 6.

3.3.2 Generalization

The network generalizes in that an input vector similar to one of the stored vectors will lead to recall of the originally stored vector, provided that distributed encoding is used. The principle by which this occurs is similar to that described for a pattern associator.

3.3.3 Graceful degradation or fault tolerance

If the synaptic weight vector w_j on each neuron (or the weight matrix) has synapses missing (e.g. during development), or loses synapses (e.g. with brain damage or ageing), then the output activation h_i (or vector of output activations h) is still reasonable, because h_i is the dot product (correlation) of r with w_j . The same argument applies if whole input axons are lost. If an output neuron is lost, then the network cannot itself compensate for this, but the next network in the brain is likely to be able to generalize or complete if its input vector has some elements missing, as would be the case if some output neurons of the autoassociation network were damaged.

3.3.4 Prototype extraction, extraction of central tendency, and noise reduction

These arise when a set of similar input pattern vectors $\{r\}$ are learned by the network. The weight vectors w_j become (or point towards) the average of that set of similar vectors. This produces 'extraction of the prototype' or 'extraction of the central tendency', and 'noise reduction'. This process can result in better recognition or recall of the prototype than of any of the exemplars, even though the effect occurs is similar to that by which it occurs in pattern general principle by which the effect occurs is similar to that by which it occurs in pattern associators. It of course only occurs if each pattern uses a distributed representation.

There has been intense debate about whether when human memories are stored, a prototype of what is to be remembered is stored, or whether all the instances or the exemplars are each stored separately so that they can be individually recalled (McClelland and Rumelhart, 1986, Ch. 17, p. 172). Evidence favouring the prototype view is that if a number of different examples of an object are shown, then humans may report that they have seen the prototype more confidently than they report having seen other exemplars, even though the prototype has never been shown (Posner and Keele, 1968; Rosch, 1975). Evidence favouring the view that exemplars are stored is that in categorization and perceptual identification tasks the responses made are often sensitive to the congruity between particular training stimuli and particular test stimuli (Brooks, 1978; Medin and Schaffer, 1978; Jacoby, 1983a,b; Whittlesea, 1983). It is of great interest that both types of phenomena can arise naturally out of distributed information storage in a neuronal network such as an autoassociator. This can be illustrated by the storage in an autoassociation memory of sets of stimuli which are all somewhat different examples of the same pattern. These can be generated, for example, by randomly altering each of the input vectors from the input stimulus. After many such randomly altered exemplars have been learned by the network, recall can be tested, and it is found that the network responds best to the original input vector, with which it has never been presented. The reason for this is that the autocorrelation components which build up in the synaptic matrix with repeated presentations of the exemplars represent the average correlation between the different elements of the vector, and this is highest for the prototype. This effect also gives the storage some noise immunity, in that variations in the input which are random noise average out, while the signal which is constant builds up with repetition.

3.3.5 Speed

The recall operation is fast on each neuron on a single iteration, because the pattern r on the axons can be applied simultaneously to the synapses w_i and the activation h_i can be accumulated in one or two time constants of the dendrite (e.g. 10–20 ms). If a simple implementation of an autoassociation net such as that described by Hopfield (1982) is simulated on a computer, then 5–15 iterations are typically necessary for completion of an incomplete input cue e . This might be taken to correspond to 50–200 ms in the brain, rather too slow for any one local network in the brain to function. However, recent work (Treves, 1993; Appendix A5) has shown that if the neurons are treated not as McCulloch-Pitt neurons which are simply 'updated' at each iteration, or cycle of time steps (and assume the active state if the threshold is exceeded), but instead are analysed and modelled as 'integrate-and-fire' neurons in real continuous time, then the network can effectively 'relax' into its recall state very rapidly, in one or two time constants of the synapses. This corresponds to perhaps 20 ms in the brain. One factor in this rapid dynamics of autoassociative networks with brain-like 'integrate-and-fire' membrane and synaptic properties is that with some spontaneous activity, some of the neurons in the network are close to threshold already before the recall cue is applied, and hence some of the neurons are very quickly pushed by the recall cue into firing, so that information starts to be exchanged very rapidly (within 1–2 ms of brain time) through the modified synapses by the neurons in the network. The progressive exchange of information starting early on within what would otherwise be thought of as an iteration period (of perhaps 20 ms, corresponding to a neuronal firing rate of 50 spikes/s), is the mechanism accounting for rapid recall in an autoassociative neuronal network that is biologically realistic in this way. Further analysis of the fast dynamics of these networks is provided in Appendix A5. The general approach applies to other networks with recurrent connections, not just autoassociators, and the fact that such networks can operate much faster than it would seem from simple models, that follow instead discrete time dynamics, is probably a major factor in enabling these networks to provide some of the building blocks of brain function.

Learning is fast, 'one-shot', in that a single presentation of an input pattern e or r enables the association between the activation of the dendrites (the post-synaptic term h_i) and the firing of the recurrent collateral axons r_i to be learned. Repeated presentation with small variations of a pattern vector is used to obtain the properties of prototype extracted from extraction of central tendency, and noise reduction, because these arise from the averaging process produced by storing very similar patterns in the network.

3.3.6 Local learning rule

The simplest learning used in autoassociation neural networks, a version of the Hebb rule, as in Eq. 3.1

$$\delta w_{ij} = k r_i r_j$$

The rule is a local learning rule in that the information required to specify the change in synaptic weight is available locally at the synapse, as it is dependent only on the presynaptic firing rate r_j available at the synaptic terminal, and the postsynaptic activation or firing r_i available on the dendrite of the neuron receiving the synapse. This makes the learning rule biologically plausible, in that the information about how to change the synaptic weight does not have to be carried to every synapse from a distant source where it is computed. As with pattern associators, since firing rates are positive quantities, a potentially interfering correlation is induced between different pattern vectors. This can be removed by subtracting the mean of the presynaptic activity from each presynaptic term, using a type of long-term depression. This can be specified as

$$\delta w_{ij} = k r_i (r_j - x) \quad (3.5)$$

where k is a learning rate constant. This learning rule includes (in proportion to r_j) increasing the synaptic weight if $(r_j - x) > 0$ (long-term potentiation), and decreasing the synaptic weight if $(r_j - x) < 0$ (heterosynaptic long-term depression). This procedure works optimally if x is the average activity $\langle r_j \rangle$ of an axon across patterns.

Evidence that a learning rule with the general form of Eq. 3.1 is implemented in at least some parts of the brain comes from studies of long-term potentiation, described in Chapter 1. One of the important potential functions of heterosynaptic long-term depression is its ability to allow in effect the average of the presynaptic activity to be subtracted from the presynaptic firing rate (see Chapter 2, Appendix A3 and Rolls and Treves, 1990).

Autoassociation networks can be trained with the error-correction or delta learning rule described in Chapter 5. Although a delta rule is less biologically plausible than a Hebb-like rule, a delta rule can help to store separately patterns that are very similar (see McClelland and Rumelhart, 1988; Hertz, Krogh and Palmer, 1991).

3.3.7 Capacity

One measure of storage capacity is to consider how many orthogonal patterns could be stored, as with pattern associators. If the patterns are orthogonal, there will be no interference between them, and the maximum number p of patterns that can be stored will be the same as the number N of output neurons (in a fully connected network). Although in practice the patterns that have to be stored will hardly be orthogonal, this is not a purely academic speculation, since it was shown how one can construct a synaptic matrix that effectively orthogonalizes any set of (linearly independent) patterns (Kohonen, 1984; see also Personnaz *et al.*, 1985; Kanter and Sompolinsky, 1987). However, this matrix cannot be learned with a local, one-shot learning rule, and therefore its interest for autoassociators in the brain is limited. The more general case of random non-orthogonal patterns, and of Hebbian learning rules, is considered next.

With non-linear neurons used in the network, the capacity can be measured in terms of the number of input patterns r (produced by the external input e , see Fig. 3.1) that can be stored in the network and recalled later whenever the net falls within their basin of attraction. The first quantitative analysis of storage capacity (Amit, Gutfreund and Sompolinsky, 1987) considered a fully connected Hopfield (1982) autoassociator model, in which units are binary

elements with an equal probability of being 'on' or 'off' in each pattern, and the number C of inputs per unit is the same as the number N of output units (actually, it is equal to $N - 1$, since a unit is taken not to connect to itself). Learning is taken to occur by clamping the desired patterns on the network and using a modified Hebb rule, in which the mean of the presynaptic and postsynaptic firings is subtracted from the firing on any one learning trial (this amounts to a covariance learning rule, and is described more fully in Appendix A4). With such fully distributed random patterns, the number of patterns that can be learned is (for C large) $p \approx 0.14C = 0.14N$, hence well below what could be achieved with orthogonal patterns or with an 'orthogonalizing' synaptic matrix. Many variations of this 'standard' autoassociator model have been analysed subsequently.

Treves and Rolls (1991) have extended this analysis to autoassociation networks which are much more biologically relevant in the following ways. First, some or many connections between the recurrent collaterals and the dendrites are missing (this is referred to as diluted connectivity, and results in a non-symmetric synaptic connection matrix in which w_{ij} does not equal w_{ji} , one of the original assumptions made in order to introduce the energy formalism of the Hopfield model). Second, the neurons need not be restricted to binary threshold neurons but can have a threshold-linear activation function (see Fig. 1.3). This enables the neurons to assume real continuously variable firing rates, which are what is found in the brain (Rolls and Tovee, 1994). Third, the representation need not be fully distributed (with half the neurons 'on', and half 'off'), but instead can have a small proportion of the neurons firing above a spontaneous rate, which is what is found in parts of the brain such as the hippocampus. Neurons are involved in memory (see Treves and Rolls, 1994, and Chapter 6). Such a representation is defined as being sparse, and the sparseness of the representation a can be measured, by extending the binary notion of the proportion of neurons that are firing, as

$$a = \frac{\sum_{i=1}^N r_i}{N} \quad (1)$$

where r_i is the firing rate of the i th neuron in the set of N neurons. Treves and Rolls (1994) have shown that such a network does operate efficiently as an autoassociative network, and can store (and recall correctly) a number of different patterns p as follows

$$p \approx \frac{C}{a \ln(1/a)} \quad (2)$$

where C is the number of synapses on the dendrites of each neuron devoted to the recurrent collaterals from other neurons in the network, and k is a factor that depends weakly on the detailed structure of the rate distribution, on the connectivity pattern, etc., but is roughly the order of 0.2–0.3.

The main factors that determine the maximum number of memories that can be stored in an autoassociative network are thus the number of connections on each neuron devoted to the recurrent collaterals, and the sparseness of the representation. For example, $C = 12000$ and $a = 0.02$, p is calculated to be approximately 36000. This storage capacity can be realized, with little interference between patterns, if the learning rule includes a form of heterosynaptic long-term depression that counter-balances the effects of associ-

long-term potentiation (Treves and Rolls, 1991; see Chapter 2 and Appendix A4). It should be noted that the number of neurons N (which is greater than C , the number of recurrent collateral inputs received by any neuron in the network from the other neurons in the network) is not a parameter which influences the number of different memories that can be stored in the network. The implication of this is that increasing the number of neurons (without increasing the number of connections per neuron) does not increase the number of different patterns that can be stored (see Appendix A4), although it may enable simpler encoding of the firing patterns, for example more orthogonal encoding, to be used.

3.3.8 Context

The environmental context in which learning occurs can be a very important factor which affects retrieval in humans and other animals. Placing the subject back into the same context in which the original learning occurred can greatly facilitate retrieval.

Context effects arise naturally in association networks if some of the activity in the network reflects the context in which the learning occurs. Retrieval is then better when that context is present, for the activity contributed by the context becomes part of the retrieval cue for the memory, increasing the correlation of the current state with what was stored. (A strategy for retrieval arises simply from this property. The strategy is to keep trying to recall as many fragments of the original memory situation, including the context, as possible, as this will provide a better cue for complete retrieval of the memory than just a single fragment.)

The very well-known effects of context in the human memory literature could arise in the simple way just described. An implication of the explanation is that context effects will be especially important at late stages of memory or information processing systems in the brain, for there information from a wide range of modalities will be mixed, and some of that information could reflect the context in which the learning takes place. One part of the brain where such effects may be strong is the hippocampus, which is implicated in the memory of recent episodes, and which receives inputs derived from most of the cortical information processing streams, including those involved in space (see Chapter 6).

3.3.9 Mixture states

If an autoassociation memory is trained on pattern vectors A , B , and $A + B$ (i.e. A and B are both included in the joint vector $A + B$; that is if the vectors are not linearly independent), then the autoassociation memory will have difficulty in learning and recalling these three memories as separate, because completion from either A or B to $A + B$ tends to occur during recall. (This is referred to as configurational learning in the animal learning literature, see e.g. Sutherland and Rudy, 1991.) This problem can be minimized by re-representing A , B , and $A + B$ in such a way that they are different vectors before they are presented to the autoassociation memory. This can be performed by recoding the input vectors to minimize overlap using, for example, a competitive network, and possibly involving expansion recoding, as described for pattern associators (see Chapter 2, Fig. 2.13). It is suggested that this is a function of the dentate granule cells in the hippocampus, which precede the CA3 recurrent collateral network (Treves and Rolls, 1992, 1994; and see Chapter 6).

3.3.10 Memory for sequences

One of the first extensions of the standard autoassociator paradigm that has been explored in the literature is the capability to store and retrieve not just individual patterns, but whole sequences of patterns. Hopfield, in the same 1982 paper, suggested that this could be achieved by adding to the standard connection weights, which associate a pattern with itself, a new, asymmetric component, that associates a pattern with the next one in the sequence. In practice this scheme does not work very well, unless the new component is made to operate on a slower time scale than the purely autoassociative component (Kleinfeld, 1986; Sompolinsky and Kanter, 1986). With two different time scales, the autoassociative component can stabilize a pattern for a while, before the heteroassociative component moves the network, as it were, into the next pattern. The heteroassociative component moves the network in the sequence is just the previous pattern in the sequence. A particular type of 'slower' operation occurs if the asymmetric component acts after a delay τ . In this case the network sweeps through the sequence, staying for a time of order τ in each pattern.

One can see how the necessary ingredient for the storage of sequences is only a minor departure from purely Hebbian learning: in fact, the (symmetric) autoassociative component of the weights can be taken to reflect the Hebbian learning of strictly simultaneous conjunctions of pre- and postsynaptic activity, whereas the (asymmetric) heteroassociative component can be implemented by Hebbian learning of each conjunction of postsynaptic activity with presynaptic activity *shifted* a time τ in the past. Both components can then be seen as resulting from a generalized Hebbian rule, which increase the weight whenever postsynaptic activity is paired with presynaptic activity occurring within a given time range, that may extend from a few hundred milliseconds in the past up to include strictly simultaneous activity. This is similar to a trace rule (see Chapter 8), which itself matches very well the observed conditions for induction of long-term potentiation, and appears entirely plausible; the learning rule necessary for learning sequences, though, is more complex than a simple trace rule in that the time-shifted conjunctions of activity that are encoded in the weights must in retrieval produce activations that are time-shifted as well (otherwise one falls back in the Hopfield (1982) proposal, which does not quite work). The synaptic weights should therefore keep separate 'traces' of what was simultaneous and what was time-shifted during the original experience, and this is not very plausible.

A series of recent investigations of the storage of temporal sequences in autoassociators is that by Levy and colleagues (Levy, Wu and Baxter, 1995; Wu, Baxter and Levy, 1996).

Another way in which a delay could be inserted in a recurrent collateral path in the brain is by inserting another cortical area in the recurrent path. This could fit in with the cortico-cortical backprojection connections described below and in Chapters 6 and 10, which would introduce some conduction delay.

3.4 Use of autoassociation networks in the brain

Because of its 'one-shot' rapid learning, and ability to complete, this type of network is well suited for episodic memory storage, in which each episode must be stored and recalled later from a fragment, and kept separate from other episodic memories. It does not take a long

time (the 'many epochs' of backpropagation networks) to train this network, because it does not have to 'discover the structure' of a problem. Instead, it stores information in the form in which it is presented to the memory, without altering the representation. An autoassociation network may be used for this function in the CA3 region of the hippocampus (see Chapter 6).

An autoassociation memory can also be used as a short term memory, in which iterative processing round the recurrent collateral loop keeps a representation active until another input cue is received. This may be used to implement many types of short term memory in the brain. For example, it may be used in the perirhinal cortex and adjacent temporal lobe cortex to implement short term visual object memory (Miyashita and Chang, 1988; Amit, 1995); in the dorsolateral prefrontal cortex to implement a short term memory for spatial responses (Goldman-Rakic, 1996); and in the frontal eye fields to implement a short term memory for where eye movements should be made in space. Such an autoassociation memory in the temporal lobe visual cortical areas may be used to implement the firing which continues for often 300 ms after a very brief (16 ms) presentation of a visual stimulus (Rolls and Tovee, 1994), and may be one way in which a short memory trace is implemented to facilitate invariant learning about visual stimuli (see Chapter 8). In all these cases, the short term memory may be implemented by the recurrent excitatory collaterals which connect nearby pyramidal cells in the cerebral cortex. The connectivity in this system, that is the probability that a neuron synapses on a nearby neuron, may be in the region of 10% (Braitenberg and Schuz, 1991; Abeles, 1991).

The recurrent connections between nearby neocortical pyramidal cells may also be important in defining the response properties of cortical cells, which may be triggered by external inputs (from for example the thalamus or a preceding cortical area), but may be considerably dependent on the synaptic connections received from nearby cortical pyramidal cells.

The cortico-cortical backprojection connectivity described in Chapters 6, 7, and 10 can be interpreted as a system which allows the forward-projecting neurons in one cortical area to be linked autoassociatively with the backprojecting neurons in the next cortical area (see Figs 6.1, 7.12, 7.13, and 10.7). This particular architecture may be especially important in constraint satisfaction (as well as recall), that is it may allow the networks in the two cortical areas to settle into a mutually consistent state. This would effectively enable information in higher cortical areas, which would include information from more divergent sources, to influence the response properties of neurons in earlier processing stages.