5 Unsupervised Learning: Autoassociative Networks and the Hippocampus

Chapters 3 and 4 described several neural networks that can be trained to associate cues, such as tone and lights, with a reinforcing stimulus, such as the airpuff unconditioned stimulus (US). By definition, a US is something qualitatively different from other cues: Whereas any detectable stimulus can become a conditioned stimulus (CS), USs are prewired to evoke reflexive behavioral responses. Thus, an airpuff US evokes a protective eyeblink, a footshock US evokes a reflexive movement of the leg away from the offending site, and a food US evokes feeding behavior in a hungry animal. In human learning, the concept of a US is often broadened to mean any stimulus that is to be predicted. Thus, in the fictitious medical diagnosis task of chapter 4 (figure 4.11), the presence of the rare disease was predicted by subjects on the basis of symptoms, and so the symptoms functioned as CSs that predicted a disease US.

What all these paradigms have in common is that there is a special input, the US, and the job of the animal, human, or computational model is to predict this US on the basis of available cues. The conditioned response is correct insofar as it predicts the US. This principle forms the basis of error-correction rules such as the Widrow-Hoff rule, embedded in the Rescorla-Wagner model. Learning occurs exactly when there is a difference between the prediction of the US (embodied by the conditioned response) and the actual US. In a sense, learning is **supervised** by a system that monitors this prediction error and adjusts associative weights accordingly.

Such error-correction systems have their uses, but—as was described in the previous chapter—they fail on paradigms such as sensory preconditioning and latent inhibition in which learning takes place in the absence of any explicit US and hence in the absence of any prediction error. Such paradigms are sometimes called **mere exposure** paradigms or **latent learning** paradigms. Learning in these paradigms is often called **unsupervised learning,** in contrast to supervised learning that is based on predicting an external reinforcement.

The dichotomy between supervised and unsupervised learning has a long history in psychology. In the first half of the twentieth century, this division defined two different schools of learning research. One school, led by Clark Hull, emphasized the study of supervised learning in which there was an explicit goal, reward, or punishment for an animal or person's behavior.[1] Rescorla and Wagner followed in this tradition in developing their model of classical conditioning.[2]

A second school of researchers, led by Edward Tolman and working at about the same time as Hull, chose to focus on unsupervised forms of learning in which animals were merely exposed to a novel environment but not trained on any particular task.[3] Tolman was especially interested in how animals learned during the exploration of new spatial environments. For example, in one study, rats were placed at the starting point of a maze and rewarded with cheese when they found their way to an endpoint in the maze.[4] The experimenters recorded how often the rats deviated from the most direct path to the goal.

As figure 5.1 shows, rats gradually made fewer and fewer errors in the maze, finding their way to the cheese reward faster and more efficiently. A second group of rats was given the same task but without the cheese. As might be expected, these rats showed no particular tendency to head toward the goal location but instead appeared to wander randomly about the maze. One might assume, therefore, that these unreinforced rats were learning little about the maze. However, after they wandered the maze for ten days, cheese was placed in the goal location starting on the eleventh day. As seen in figure 5.1, these rats made a veritable beeline for the goal: On day 13, they reached the goal faster, on average, than the rats that had been given the food reward all along. Tolman interpreted this result as implying that although the unreinforced rats appeared to be wandering aimlessly through the maze during the first part of the study, they were in fact actually exploring and learning about their environment. Later, when challenged to navigate the maze, the rats used this prior learning to guide their path.

From these and similar experiments, Tolman concluded that animals can and do naturally learn about their environment even when they are not explicitly rewarded for doing so. Tolman argued that the rats in figure 5.1 formed **cognitive maps**—mental models of their environment—during exposure to a maze.

Interestingly, it is exactly this kind of unreinforced learning that seems to be chiefly disrupted by hippocampal-region damage in animals. Recall from chapter 2 that rats with hippocampal-region damage are unable to learn the location of a submerged platform in a milky pool and do not show behaviors

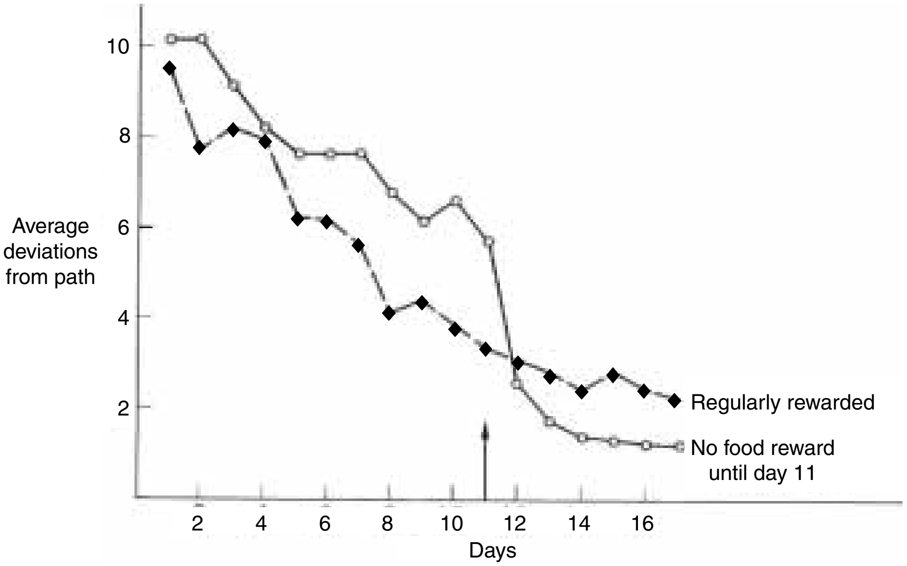


Figure 5.1 Rats placed at the start location in a maze, with a food reward regularly available in a goal location, learn to run directly to the goal, as indicated by fewer deviations from the most direct path. A second group of rats are placed in the maze, allowed to run in the maze without food for ten days, and then placed in the maze with food available from day 11 on; these rats learn very quickly to run directly to the food, quickly outperforming even the rats who received food reward all along. This implies that during the first ten unrewarded days, the rats were exploring and learning about the maze environment; later, when a food reward was available, they were able to use that information to find food efficiently. (Based on Hilgard & Bower, 1975, p. 135, Figure 5.3, adapted from Tolman & Honzik, 1930.)

such as sensory preconditioning and latent learning that depend on learning about cues presented in the absence of a reinforcing US. Thus, to a first approximation, it would seem that unsupervised "Tolmanian" (CS-CS) learning depends on the hippocampal region, while supervised "Hullian" (CS-US) learning does not. To the extent that this mapping holds true, it would suggest that the error-correction networks from chapter 3 may be better descriptions of learning in hippocampal-lesioned animals than in intact animals.

To take this simple mapping a step further, if error-correcting networks can capture supervised (CS-US) learning of the kind seen in hippocampal-lesioned animals, what kind of network architecture is capable of the unsupervised (CS-CS) learning that is seen in animals with an intact hippocampus? There do, in fact, exist network architectures that are capable of the kinds of unsupervised learning that Tolman described. These networks are called **autoassociative networks** and are described below. The autoassociative networks are of particular interest, not only because they capture some features of hippocampal-dependent learning, but also because there is a long tradition in neuroscience of using them as models of how certain circuits within the hippocampus operate.

5.1 AUTOASSOCIATIVE NETWORKS

In an influential 1949 book, Donald Hebb proposed that the brain stores information via a simple rule: *If one cell (or neuron) A connects to a second cell B, and if the two cells are repeatedly active at the same time, then the connection between them is strengthened, so future activity in A is more likely to cause activation in B*.[5] This rule is often called **Hebb's rule,** and the learning it embodies is called **Hebbian learning.**

In contrast to the supervised learning characterized by the Widrow-Hoff learning rule, Hebbian learning is unsupervised, meaning that it does not depend on a special teaching signal, such as a reinforcing US. Weight change is automatic in Hebb's rule and depends only on conjoint activity in pairs of nodes.

Hebb's rule was originally proposed as a theory of how large assemblies of neurons in the brain operate. Later neurophysiological studies verified that Hebbian learning does occur in the brain: Synapses between coactive neurons do grow stronger. This process is called **long-term potentiation (LTP),**[6] because the synaptic change (potentiation) has been observed to last for weeks in the laboratory and may last longer in living animals. This synaptic process (there are several variations) has been observed throughout the brain[7] and is currently believed to underlie many kinds of changes in the brain.[8]

Figure 5.2A shows a simple example of an autoassociative network,[9] trained according to Hebb's rule. Eight nodes are shown, and each has a weak connection to all the others. Each node is *both* an input node and an output node, meaning that it receives input from external sources, while its activation level is part of the final output produced by the network. External inputs arrive and activate some subset of nodes in the network (darkened circles in figure 5.2A); other nodes remain inactive. This pattern of active and inactive nodes represents knowledge to be stored by the network. According to Hebb's rule, connections between coactive nodes are strengthened (figure 5.2B). This is usually done in one massive step, by simply setting all the relevant weights to 1.0. At this point, the pattern is **stored** in the network. A different input pattern, which activates different nodes (figure 5.2C), could be stored in the same way (figure 5.2D).

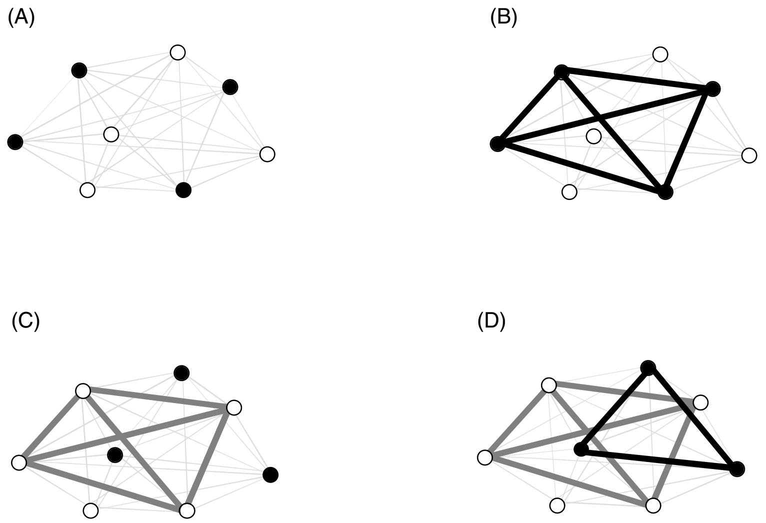


Figure 5.2 Simplified example of an autoassociative network, consisting of a group of nodes (circles). (A) External inputs (not shown) evoke a pattern of activity over the network, resulting in the activation of a subset of nodes (solid circles). (B) Weighted connections develop between coactive nodes; the pattern is said to be stored. (C) Later, a different pattern is presented that activates a different subset of nodes (solid circles). (D) Weights between these coactive nodes are strengthened, and the new pattern is stored.

The ability to remember patterns in this way is one mechanism by which the brain might store memories of past events. Once a pattern has been stored, activation in any one node will produce activation in the other nodes of a pattern. Thus, even after the external input is turned off, the pattern (stored as an assembly of active nodes) remains for some time, until activation gradually dies away. In this way, superficially unrelated but temporally coincident stimuli can be bound together into a unified memory. For example, an episodic memory of meeting a celebrity might include not only the gist of the conversation with that person, but also details of the cocktail party where the meeting occurred, such as the time and place and who else was present. Each of these details could be represented as a node (or a group of nodes) that could then be bound together by synaptic changes between these coactive nodes.

Of course, a stored pattern is no use unless it can be retrieved later. And this leads to the most interesting property of an autoassociative network: *Given a partial or incomplete version of a stored memory, an autoassociative network can retrieve the entire stored pattern*. This is called **pattern completion,** and figure 5.3 shows an example. The network has already stored two patterns, represented

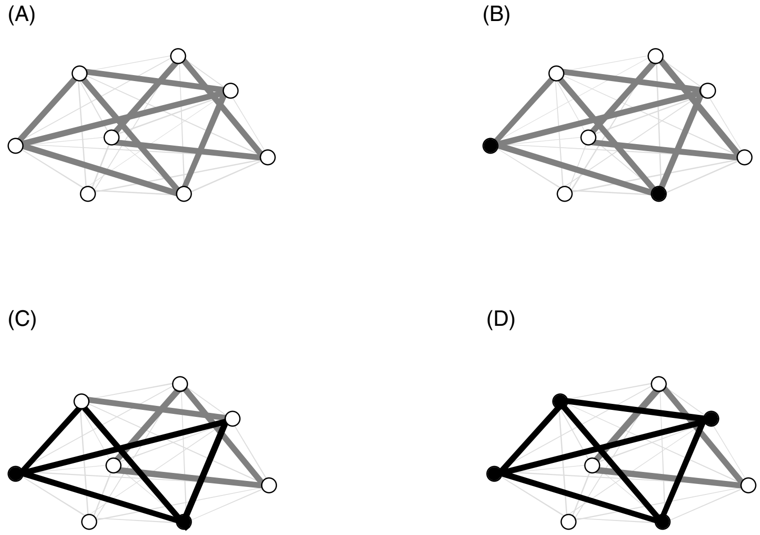


Figure 5.3 Pattern completion in an autoassociative network. (A) The network has stored two patterns by strengthening the weights between nodes that are coactive in each pattern. (B) A partial version of one pattern is presented, which activates a subset of the nodes (solid circles). (C) Activation flows from active nodes along previously strengthened connections (heavy lines), (D) and the entire pattern is retrieved.

by strong connections between the nodes that were coactive (figure 5.3A). Now, a partial version of one stored pattern is presented, which activates only a few of the nodes in the stored pattern (figure 5.3B). Activation then spreads along the previously strengthened connections (figure 5.3C), activating the remaining nodes (figure 5.3D). Finally, the entire pattern is reinstated.

In the previous example of remembering a celebrity, pattern completion means that, given a mention of the individual, spreading activation will activate nodes corresponding to the details of the party where the celebrity was encountered—retrieving the complete memory from a partial cue.

A variant of this process is **pattern recognition,** the ability to take an arbitrary input and retrieve the stored pattern that is most similar to that input. In figure 5.4A, the network has been trained on the same two patterns as in the earlier example. Next, in figure 5.4B, the external inputs present a new pattern that overlaps partially with one of the stored patterns. Activation will spread along weighted connections to activate the other nodes in the stored pattern (figure 5.4C). At the same time, lack of such activity will mean that the extra node that is incorrectly activated will eventually become quiet. In the end, the stored pattern is correctly retrieved (figure 5.4D). The network has "recognized" its input as a distorted version of the stored pattern.

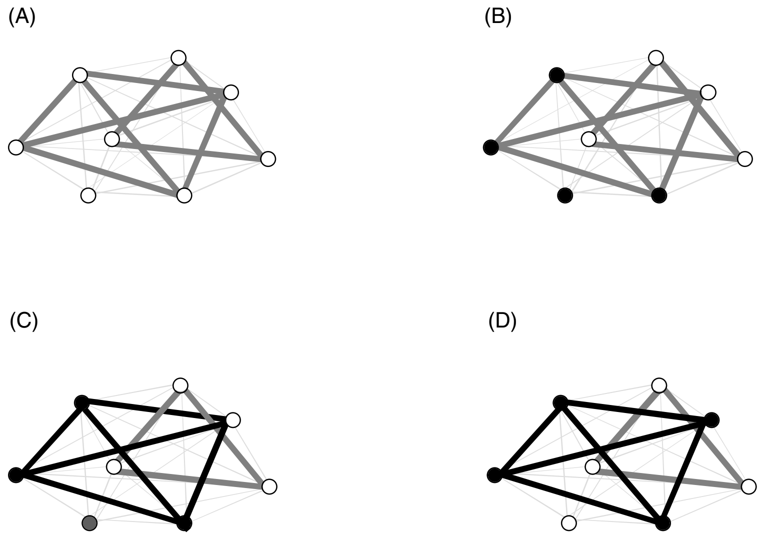


Figure 5.4 Pattern recognition in an autoassociative network. (A) The network has stored two patterns by strengthening the weights between nodes that are coactive in each pattern. (B) External inputs activate nodes (solid circles), some of which are part of a previously stored pattern and some of which are not. (C) Activation spreads along previously strengthened pattern, activating nodes that are part of the stored pattern. Other nodes, which do not receive this input, gradually become inactive until (D) the stored pattern is retrieved. At this point, the network has recognized the input in (B) as a degraded version of the stored pattern retrieved in (D).

In addition to serving as models of memory for events,[10] the twin features of pattern completion and pattern recognition have led to many engineering applications of autoassociative networks. These applications range from interpreting sonar signals to guiding robot navigation to optimizing how tasks are assigned to cooperating units.[11]

5.2 HIPPOCAMPAL ANATOMY AND AUTOASSOCIATION

Traditionally, most theories that have tried to map from hippocampal anatomy to behavioral function have focused on a particular subfield within the hippocampus: field CA3. As shown in figure 5.5, CA3 is roughly the part of the hippocampus enclosed by the dentate gyrus. Hippocampal field CA1 lies next to CA3 and merges into the subiculum and dentate gyrus.

Figure 5.6 shows a schematic of information flow into and out of CA3. One primary input comes from entorhinal cortex, carrying highly processed information about all kinds of sensory input. This pathway is called the **perforant path,** since its fibers physically perforate the dentate gyrus to reach

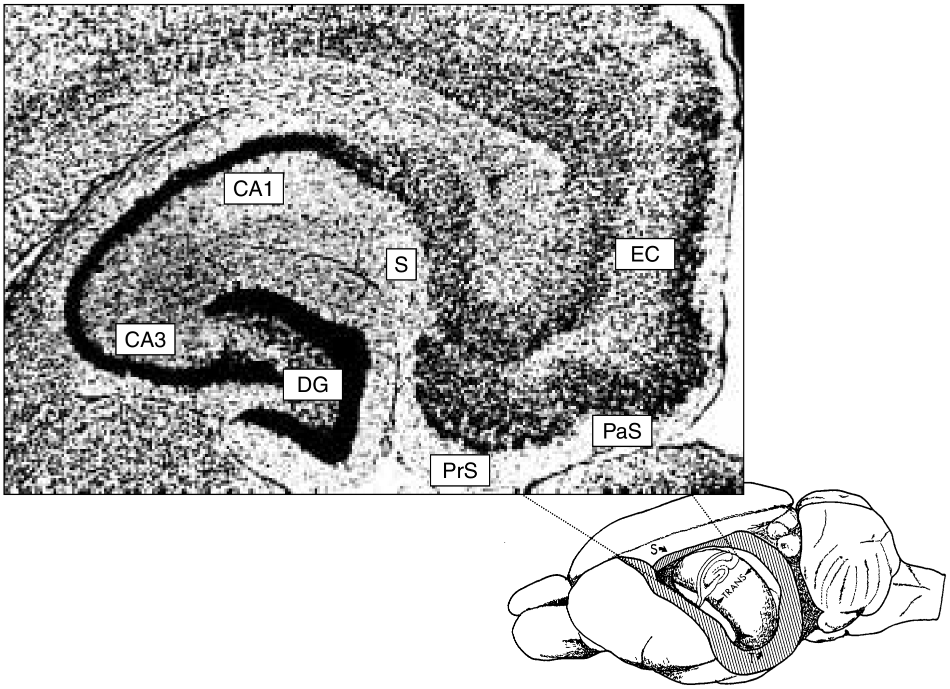


Figure 5.5 Lower right: Drawing of the rat brain, with surface removed to show the position of the hippocampus; the cutaway view shows the internal organization of the hippocampal region. Top left: Photomicrograph of a cutaway section through the rat hippocampus. The hippocampus (including fields CA3 and CA1) is visible as a "C"-shaped line of neurons; the dentate gyrus (DG) is an interlocking "C"-shaped line of cells. Adjacent to hippocampal field CA1 is the subicular complex (including S = subiculum, PrS = presubiculum, PaS = parasubiculum) and entorhinal cortex (EC). (Adapted from Amaral & Witter, 1989, Figures 1 and 2.)

CA3. A secondary path from the entorhinal cortex synapses in dentate gyrus before proceeding on to CA3. The connections from dentate gyrus to CA3 are called **mossy fibers,** and they make sparse, large, and presumably very powerful synapses onto CA3 neurons. CA3 neurons process this information and send their outputs on to hippocampal field CA1 and from there out of the hippocampus.

One of the most striking features of CA3 anatomy is the high degree of **internal recurrency,** meaning that CA3 neurons send axons not only out of CA3, but also back to synapse on other CA3 neurons. Such feedback is a general principle throughout the brain, but it is dramatically heightened in CA3. For example, each CA3 pyramidal neuron in the rat may receive about 4,000 synapses from entorhinal inputs but up to about 12,000 synapses from other CA3 cells.[12] This means that each CA3 pyramidal neuron

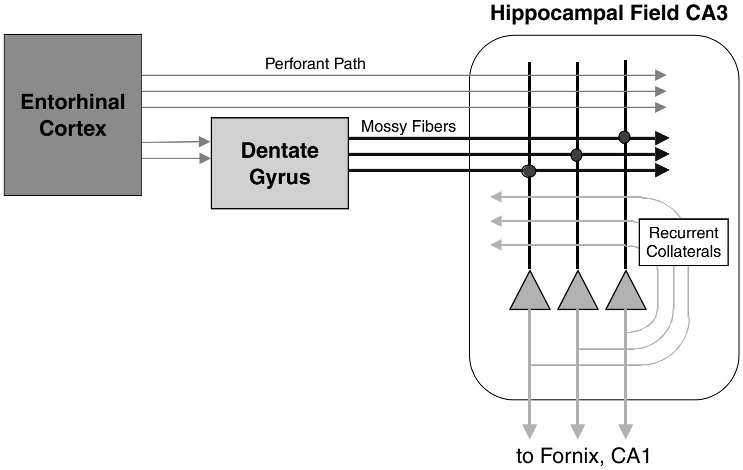


Figure 5.6 Schematic of information pathways into and within hippocampal field CA3. Highly processed, multimodal sensory information from entorhinal cortex travels via the perforant path to contact pyramidal neurons in CA3. There is also an indirect path from entorhinal cortex via the dentate gyrus (the mossy fiber path). CA3 pyramidal neurons have a high degree of internal recurrence, meaning that axons have collaterals that travel back to contact other CA3 neurons, as well as sending information out of CA3 to the fornix, CA1, and elsewhere.

receives inputs from about 4% of all other CA3 pyramidal neurons. This is nowhere near the 100% recurrency that is assumed in the simplified network models of figures 4.2 through 4.4, in which every node is connected to every other node. However, this 4% recurrency is orders of magnitude higher than the degree of internal recurrency that is observed elsewhere in the brain. Furthermore, the connections between CA3 neurons are modifiable by LTP in much the same manner as Hebb proposed. In fact, LTP was originally discovered by researchers studying synapses in the hippocampal region.[13]

Given these anatomical and physiological characteristics, it seems logical to ask whether CA3 could function as an autoassociative network. One of the earliest and most influential models of hippocampus, published by David Marr in 1971, proposed exactly that.[14] In its simplest phrasing, Marr's model assumes that CA3 pyramidal neurons form an autoassociative network in which external inputs (from entorhinal cortex and dentate gyrus) activate a subset of CA3 neurons. Recurrent collaterals between coactive nodes are strengthened, storing the pattern. Since Marr's original publication, a wealth of new empirical data has shown that some additional aspects of his model were incomplete or incorrect.[15] Nonetheless, many of the basic ideas underlying Marr's model have withstood continuing empirical and theoretical analysis and are implicit in most modern models of hippocampus.[16]

Storage Versus Retrieval

One important requirement for an autoassociator, implicit in the above discussion, is that the network be able to operate in two distinct modes: a storage mode and a recall mode. Remember the example of figure 5.2D, in which one pattern has been stored in the network and a second pattern is presented. How does the network "know" whether this second pattern represents new information to be stored (as in the example of figures 5.2C and 5.2D) or a distorted version of an old pattern to be retrieved (as in figures 5.3 and 5.4)?

The simplest assumption is that there is an external input, a "teacher," that guides the network into either a storage state or a recall state. This may not be as unlikely as it sounds; recent evidence has suggested that areas of the brain may be flooded with chemicals (called **neuromodulators**) that encourage the brain either to store new information or to retrieve familiar information. Chapter 10 discusses this idea in more detail, with particular reference to the neuromodulator **acetylcholine.**

However, the anatomy of the hippocampus suggests a second (possibly complementary) way to guide the network into storage or recall mode. Figure 5.6 illustrated the two principal information pathways into hippocampal field CA3: the direct perforant path from entorhinal cortex and the indirect path via dentate gyrus, which terminates in mossy fiber synapses onto CA3 neurons. These mossy fiber synapses are very sparse: In the rat, each CA3 pyramidal neuron receives only about 50 mossy fiber synapses,[17] a mere fraction of the inputs from the perforant path or from recurrent collaterals. However, these mossy fiber synapses are physically much larger and stronger than the other inputs: Whereas input from hundreds or thousands of entorhinal inputs may be required to activate a CA3 pyramidal neuron, coincident activity on a relatively small number of mossy fiber inputs may be sufficient to activate the CA3 neuron.[18]

The implication is that mossy fiber inputs may be strong enough to act as teaching inputs, causing pattern storage. In the absence of mossy fiber activation, the network may perform pattern retrieval and pattern completion, as schematized in figure 5.7. When a new pattern is presented for storage, the strong mossy fiber inputs cause the CA3 neurons that they contact to become strongly active (figure 5.7A). According to Hebb's rule, strong conjoint activity in two neurons causes strengthening of the connections (synapses) between them (figure 5.7B). At this point, the pattern is stored. Later, a partial version of the input is presented along weaker perforant path inputs; this produces partial activation in some of the same CA3 neurons (figure 5.7C).

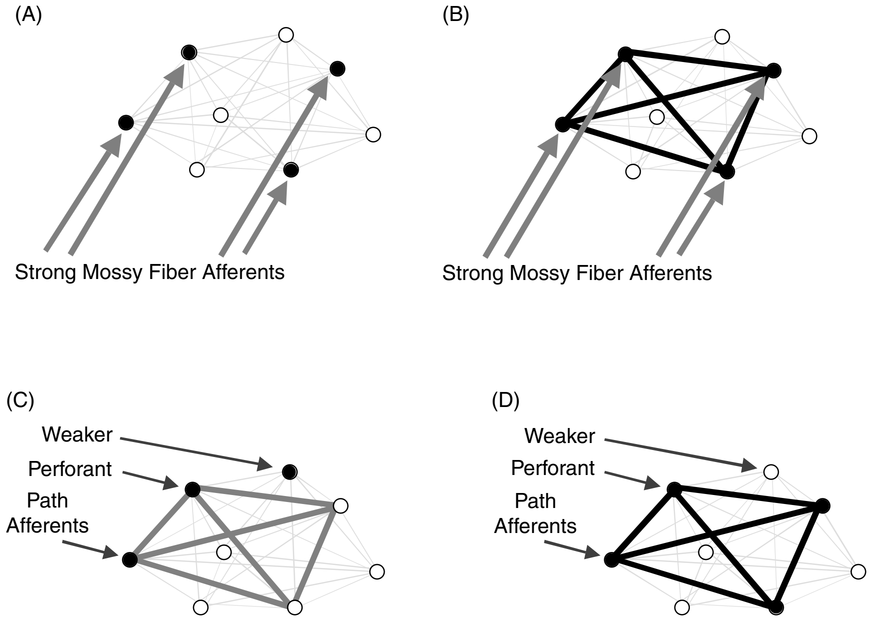


Figure 5.7 Possible implementation of autoassociation in CA3 circuitry. (A) One set of inputs, the strong mossy fibers from dentate gyrus, activate a set of CA3 neurons. (B) In the presence of this strong input, plasticity occurs along connections (synapses) between coactive neurons. (C) Another set of inputs, the perforant path afferents from entorhinal cortex, make relatively weaker synapses onto CA3 neurons, activating a subset of CA3 neurons. (D) This weaker input is enough to spread along previously strengthened connections, retrieving a stored pattern, but not enough to maintain activity in neurons that are not part of the pattern, nor enough to cause new plasticity. Thus, storage occurs when strong mossy fiber inputs are present; pattern retrieval (including pattern completion and pattern recognition) occurs when weaker perforant path inputs are present.

Activation spreads along recurrent connections that were previously strengthened—enough to activate some additional CA3 neurons but not enough to cause additional synaptic plasticity, which requires very strong conjoint activation (figure 5.7D). As a result, the stored pattern is recalled, but no new learning takes place.

This method for implementing an autoassociator in the brain relies on the fact that there exist inputs, specifically mossy fibers, that make unusually strong synapses on target neurons. Such a situation may be the exception rather than the rule, although as we noted in chapter 3, climbing fibers in the cerebellum may share a similar function: forcing cerebellar Purkinje cells to become active and guiding learning.

Capacity, Consolidation, and Catastrophic Interference

One problem with autoassociative networks is that they have very limited capacity; that is, they can store only a small number of unrelated patterns before new information begins to overwrite the old. For example, suppose one memory incorporates nodes A, B, C, and D and creates strong associations between them, such that activation of a subset will spread to retrieve the entire memory (figure 5.8A). Now, suppose a second pattern is presented that incorporates nodes A, B, E, and F and the associations between these nodes are strengthened (figure 5.8B). If the network is presented with a partial memory consisting of E and F (figure 5.8C), activation will spread to A and B (figure 5.8D) and from there to C and D (figure 5.8E), thereby "retrieving" a

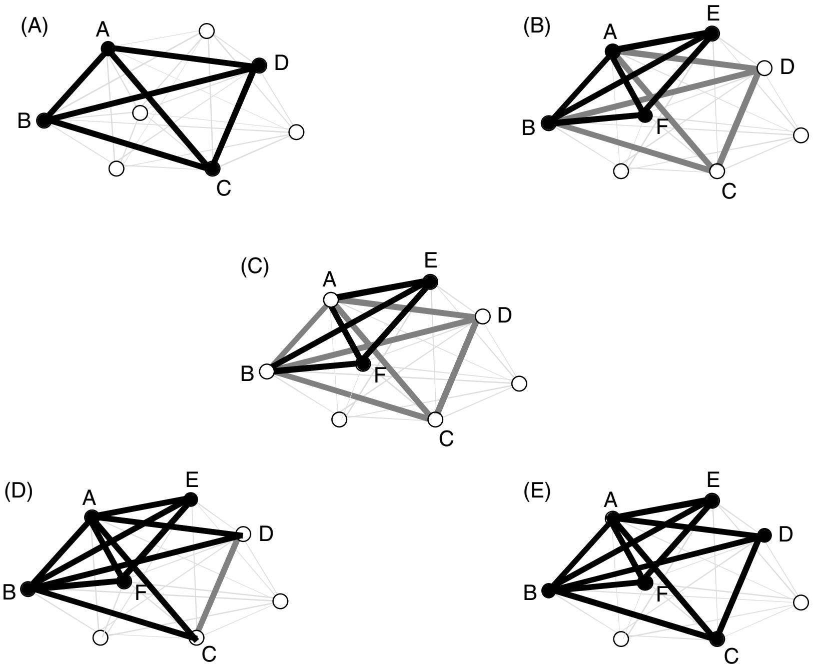


Figure 5.8 Interference in an autoassociative network. (A) One pattern has already been stored in the network by weighting connections between coactive nodes (nodes A, B, C, and D). (B) A new set of inputs is presented to the network, activating a new subset of nodes that partially overlaps with the previous pattern (nodes A, B, E, and F). Connections between coactive nodes are strengthened, and the new pattern is stored. (C) A partial pattern is presented (nodes E and F) that overlaps partially with the second stored pattern. (D) Activity spreads along previously strengthened connections to activate nodes A and B and from there (E) spreads to activate nodes C and D, thus "retrieving" a complex pattern that was never stored.

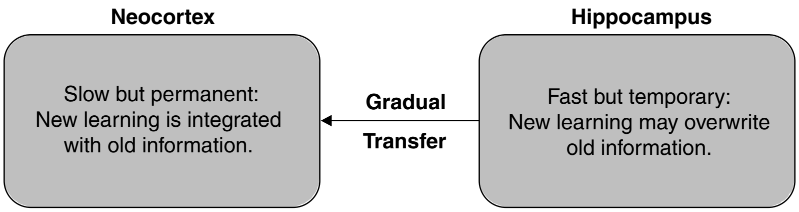


Figure 5.9 General format of many connectionist models of amnesia. The cortex is assumed to be a large-capacity, permanent store for memory associations and to be able to integrate new information with old associations. However, learning is assumed to be slow and possibly to require several iterated presentations. The hippocampus is assumed to be capable of storing memory within as little as a single exposure, but older memories are liable to be overwritten by newer ones. The hippocampus is thus only a temporary store. Over some period of time (known as the consolidation period), memories are integrated into cortex and become independent of the hippocampus.

complex pattern that was never stored! Such **interference** or confusion among stored patterns is a fundamental limitation of autoassociative networks.

Marr noted this potential for interference in autoassociative networks, and he made his second important contribution to hippocampal modeling by suggesting a possible solution.[19] He proposed that memory consisted of two separate but interacting modules (figure 5.9). One, which he located in cortex, was assumed to be a large-capacity, permanent memory store. The second, which he located in hippocampus, was a limited-capacity, limited-duration store. The hippocampal network was conceptualized as an autoassociative network. In brief, the general idea was that a stimulus would enter via the sensory systems and eventually lead to activation of cells in the cortex and hippocampus. The hippocampus would quickly strengthen connections between coactive cells, capturing the memory. Over some period of time, this information would be retrieved and stored in the cortex in such a way as not to interfere with other, earlier information. Eventually, the memory would become independent of the hippocampus, so that when new information overwrote the hippocampal cell assembly, the cortical copy would be safe.

Marr's characterization of the hippocampus as a temporary buffer for memories before they are consolidated into long-term memory provides a compelling account of three key aspects of anterograde amnesia. First, hippocampal damage results in a permanent closing of the gateway for acquiring new memories. Second, hippocampal damage does not disrupt memories that have previously been stored in the cortex. Third, since there is a consolidation period, during which memories are transferred from hippocampal to cortical storage, hippocampal damage may result in time-limited retrograde amnesia for recently stored memories.

Many connectionist models of amnesia have followed Marr in assuming this kind of fast, temporary hippocampal storage and slow, permanent cortical storage.[20] Many nonconnectionist models and theories assume this kind of organization as well.[21]

5.3 AUTOENCODERS: AUTOASSOCIATION WITH REPRESENTATION

Autoassociative models of hippocampus have considerable power for describing the hippocampal region's role in episodic memory formation. However, while autoassociative networks may be a good model of episodic learning and pattern completion—essential components of memory, to be sure—they have little to say about other, equally important kinds of learning. For example, they do not allow error-correction learning, nor do they allow the formation of new representations, such as the multilayer networks of chapter 4.

Fortunately, there is an elaboration on autoassociative networks that combines the latent learning and pattern completion abilities of autoassociators with the error-correction and representational power of multilayered networks. This kind of network is called an **autoencoder.**

A New Interpretation of Autoassociators

So far in this chapter, we have illustrated autoassociative networks as amorphous clusters, with connections drawn between individual nodes. Drawn in this way, they look very different from the associative networks that we presented in earlier chapters.

For example, figure 5.10A shows an autoassociator that interconnects nodes representing light, tone, odor, and texture cues. Each node is connected to each of the other nodes. Each node functions as both an input node and an output node: receiving activation from sources outside the network and producing output that is taken as the network's response. The same network can therefore be redrawn as in figure 5.10B. Here, each cue is represented by an input and an output node. Each input cue activates one input node, which in turn activates a corresponding output node. Input nodes also have modifiable weighted connections to all the other output nodes. The pattern of activation at the output layer is taken as the output of the network.

Here is how the autoassociator works, drawn this way. A pattern is presented for storage and activates several nodes at the input level—for example, light and odor but not tone or texture (figure 5.11A). Each input node causes activity in the corresponding output-level node; that is, the output nodes for light and odor are activated (figure 5.11B). Connections between

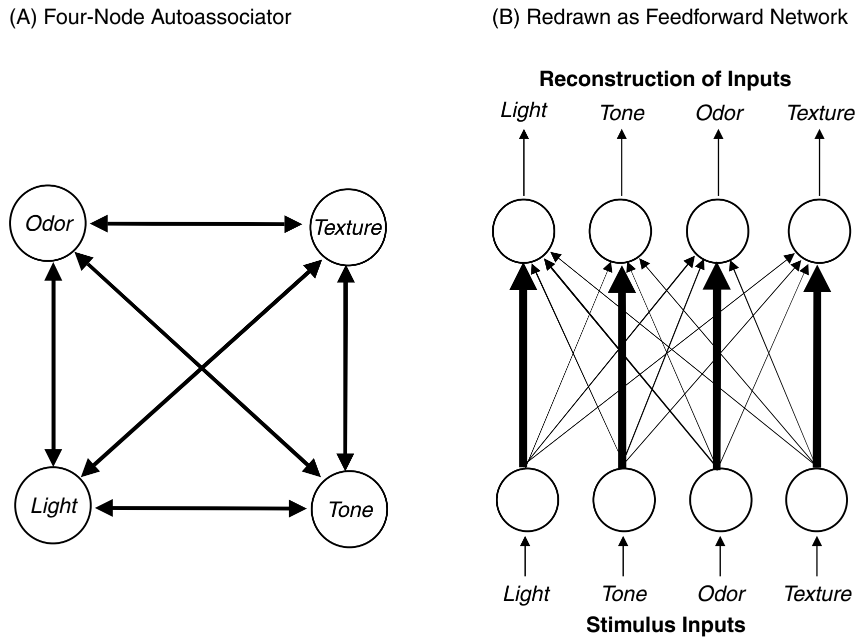


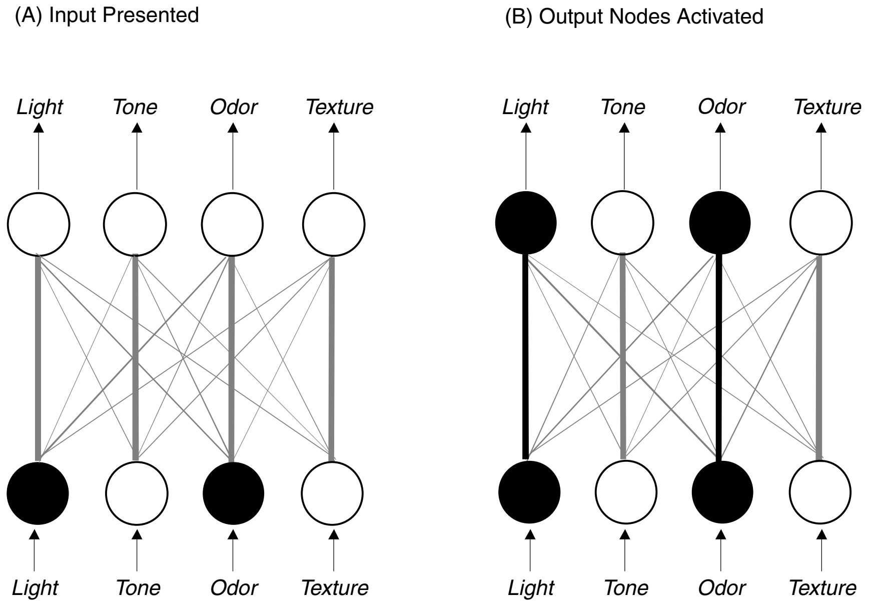
Figure 5.10 (A) An autoassociator, containing nodes representing four cues. Each node is connected to all the other nodes, and each node functions as both an input node and an output node. (B) The same autoassociator can be redrawn with separate input and output nodes for each cue. There is a strong, nonmodifiable weight from each input node to the corresponding output node. Thus, activation of the light input node causes activation of the light output node. There are also modifiable connections from each input node to each output node, so weights between coactive cues can be strengthened.

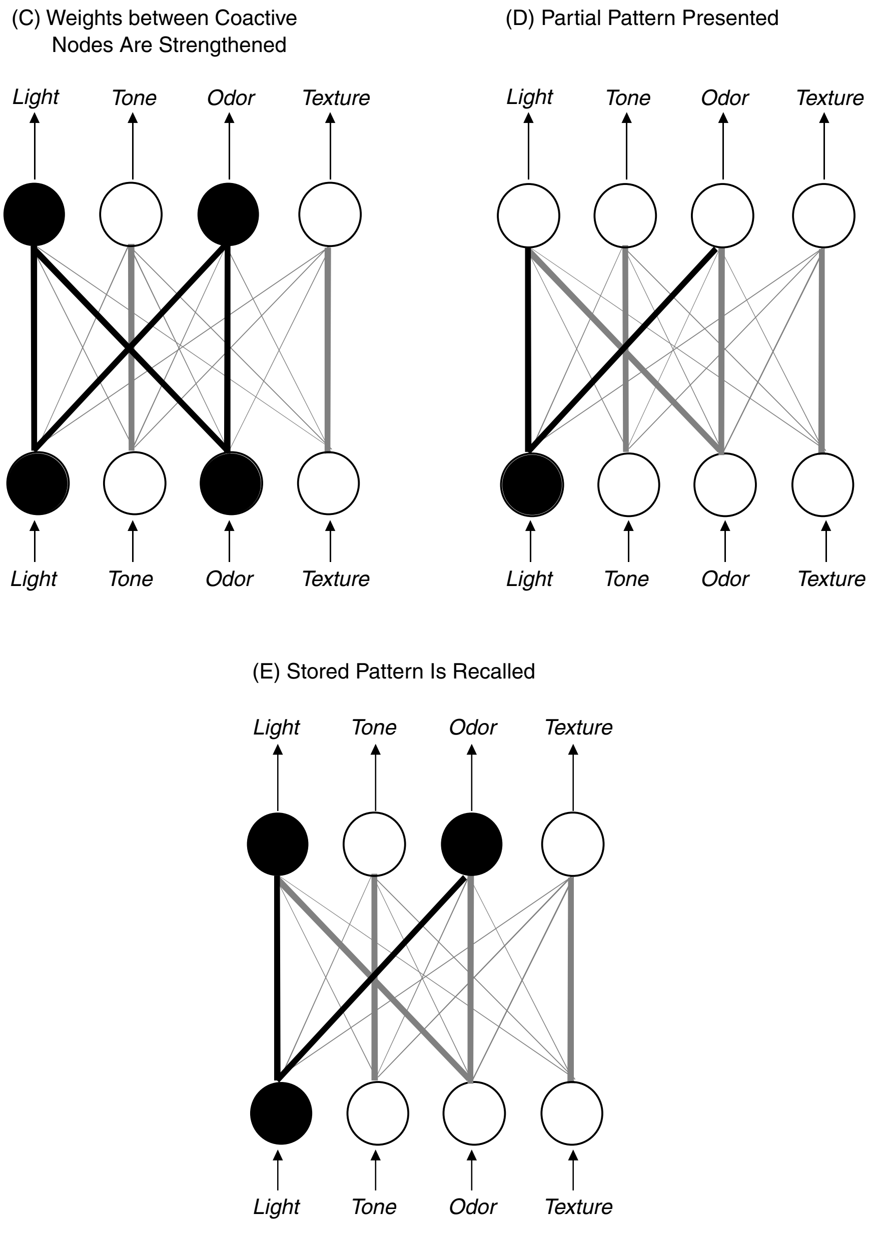
coactive nodes are strengthened. This means that connection weights between the light input node and the odor output node and those between the odor input node and the light output node are strengthened (figure 5.11C). At this point, the pattern is stored. In effect, the network has learned to associate an input pattern with an output that is the same pattern. Now it may be clear why these networks are called autoassociators: Rather than associating an input pattern with an arbitrary output (such as a reinforcing US), they associate a pattern with itself.

Later, a partial version of the stored pattern may be presented, such as light only (figure 5.11D). The inputs activate their corresponding outputs but also activate those output nodes to which they have previously strengthened connections (figure 5.11F): The output nodes corresponding to light and odor are activated, and the stored pattern is reconstructed at the output level.

This might seem like a more complicated method to picture an autoassociator than the drawings of section 5.2, but it has an important advantage: When it is drawn in this way, there is a clear difference between what portions of the pattern are presented as external input and what portions are reconstructed by the network. In figure 5.11E, the difference between the input activations and the output activations consists of elements of the pattern that the network has reconstructed. The degree of difference between the original input pattern and the output pattern produced by the network is a measure of the error in the reconstruction. If the error is high enough, this implies that the new pattern may be one that has not been seen before. As such, this reconstruction error in an autoassociative network can also be viewed as a measure of the **novelty** of the input pattern: For a familiar (previously stored) pattern, input and output will match perfectly; for a novel (never-stored) pattern, input and output will deviate; for a pattern that contains elements of a previously stored pattern, the input and output may be similar but not identical.

Cast in this way, an autoassociative network can capture some features of latent learning. In the example of figure 5.11, two stimuli (light and odor) were always presented together; strong weights developed between the input and output nodes to reflect this correlation. Later, when the light was presented alone (figure 5.11D), the network output reflected expectation that the odor should also occur (figure 5.11E). Thus, without any explicit reinforcement (i.e., US), the network learned which stimuli reliably co-occur and which (tone and texture) do not.





(C) Weights between Coactive Nodes Are Strengthened

(D) Partial Pattern Presented

(E) Stored Pattern Is Recalled

Figure 5.11 The two-layer autoassociator in action. (A) Input is presented by activating a subset of the input-layer nodes (e.g., light and odor). (B) This activates the corresponding output nodes (i.e., light and odor). (C) The weights between coactive nodes are strengthened (from light input to odor output and from odor input to light output). (D) Later, a partial version of the stored pattern is presented as input (e.g., light only). (E) Activation spreads along previously strengthened connections to retrieve the stored pattern (i.e., light and odor).

Autoencoders: Multilayer Autoassociators

A second reason for drawing autoassociators with two layers of nodes is that it is fairly easy to see how to extend the network to include an additional, internal node layer between the input and output nodes (figure 5.12). Because the internal-layer representation can be thought of as a recoding of the input pattern, this type of network is sometimes called an **autoencoder.**[22]

Like the autoassociators, the autoencoder in figure 5.12 is trained to reconstruct the input pattern on the output nodes. When the output faithfully matches the input, the autoencoder has stored the pattern. When presented with a partial or distorted version of a stored pattern, the autoencoder will reconstruct the stored pattern on its output nodes. When a novel pattern is presented, the discrepancy between input and output patterns gives a measure of this novelty.

However, just as the Widrow-Hoff rule was insufficient to train a multilayer associative network in chapter 4, Hebbian learning is insufficient to train a multilayer autoencoder—and for exactly the same reason: There is no *a priori* way to determine activations in the internal-node layer so that the

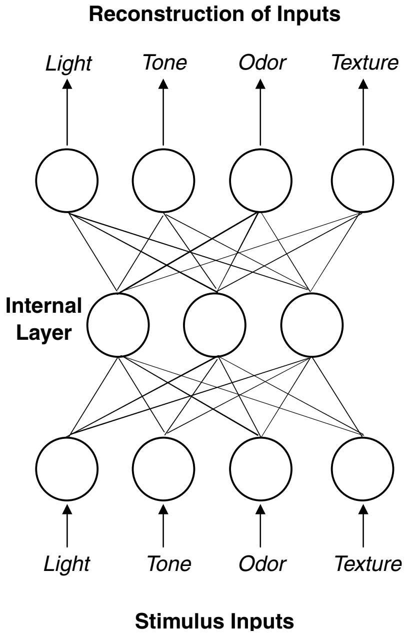


Figure 5.12 An autoencoder is an autoassociative network with an internal node layer.

outputs reconstruct the inputs faithfully. Without knowing these desired activations, there is no way to set the lower layer of weights.

However, once again, error backpropagation provides one way in which this kind of network could be trained. We know the desired responses of the output-layer nodes; they should simply replicate the pattern of activations across the input layer nodes. Knowing this, we can use one application of error-correction learning to train the upper layer of weights. The error at the output layer is backpropagated and distributed among the internal-layer nodes, giving an estimate of the desired activations there. Now, a second application of error-correction learning can be used to train the lower layer of weights. (For details on the error backpropagation algorithm, refer to MathBox 4.1.) With repeated presentations of an input pattern, the output nodes come to produce the desired responses: an accurate reconstruction of the input pattern.

Usually, autoencoder networks are drawn with the internal layer containing fewer nodes than either the input or output layer, as in figure 5.12. In this case, the autoencoder's task is complicated: In figure 5.12, the network has to accept four stimulus inputs and produce four outputs, but it has only three internal-layer nodes to encode this information. The only way to accomplish this task is if the network can identify regularities and redundancies in the input. For example, in figure 5.11, we discussed an experiment in which light and odor always co-occur. In this case, the autoencoder could make use of these statistical regularities: It would need only one internal-layer node to represent the joint occurrence of light and odor, leaving one node each to represent tone and texture (figure 5.13A). The representations of light and odor have been **compressed,** or made more similar. Later, when the light is presented alone, the network performs pattern completion and activates the output node corresponding to odor as well (figure 5.13B).

The general principle of compression, illustrated in figure 5.13, is analogous to what happens when a computer file is compressed for saving as a smaller file. Information is not lost; rather, it is encoded in a more efficient manner that requires less disk space, analogous to using fewer internal nodes in a network. Similar schemes are also used to compress speech and other sounds for efficient transmission over phone lines.

Predictive Autoencoders

So far, we have shown that autoassociative networks can learn about stimulus relationships through mere exposure to stimuli. The multilayer autoencoders are autoassociative networks that form internal-layer representations that

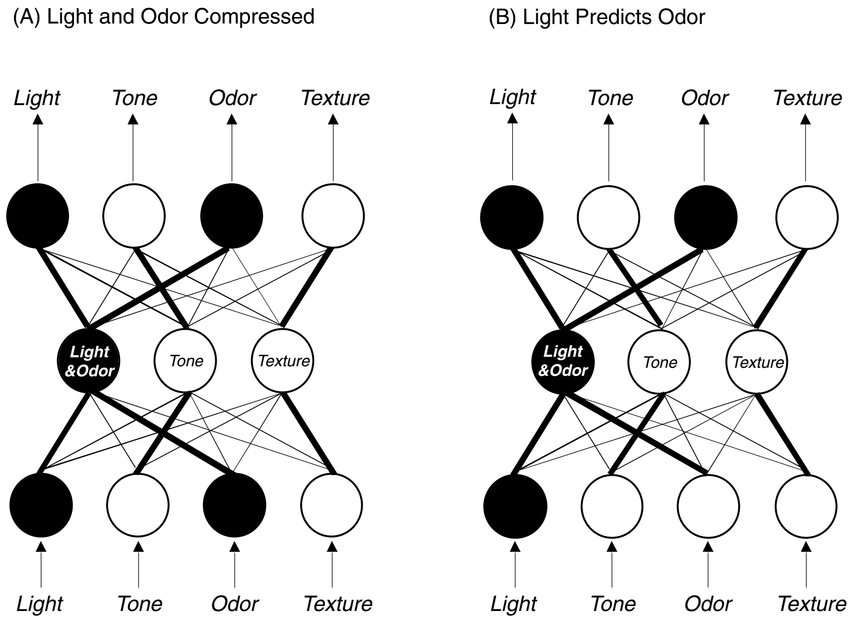


Figure 5.13 The autoencoder compresses the representations of redundant (co-occurring) stimuli in its internal layer. (A) For example, if light and odor always co-occur, one internal-layer node may come to encode both. Weights from this node to the outputs ensure that the light and odor output nodes are active whenever the corresponding inputs appear. Other internal-layer nodes encode the presence of tone and texture. The figure shows a local representation; in reality, the representation of a stimulus may be distributed over several nodes. (B) Later, if light is presented alone, the compressed internal-layer representation activates the output nodes corresponding to both light and odor.

compress stimulus redundancies while preserving nonredundant information. There is still one more important feature of learning: learning that combinations of cues predict the US reinforcement and generating an appropriate conditioned response. With one more small variation, this can be achieved.

Figure 5.14 shows a **predictive autoencoder.**[23] This is an autoencoder with one or more additional output nodes trained to predict the reinforcement. Remember that reinforcement is itself just another kind of stimulus. An airpuff US, for example, is felt as a somatosensory (touch) stimulation on the sensitive cornea; a footshock US activates pain receptors in the foot; and so on. Thus, if a tone CS reliably predicts an airpuff US, this really means that the airpuff US is a stimulus that normally occurs in the presence of the tone stimulus. In effect, tone and airpuff are elements of a single pattern that the network should learn to reproduce.

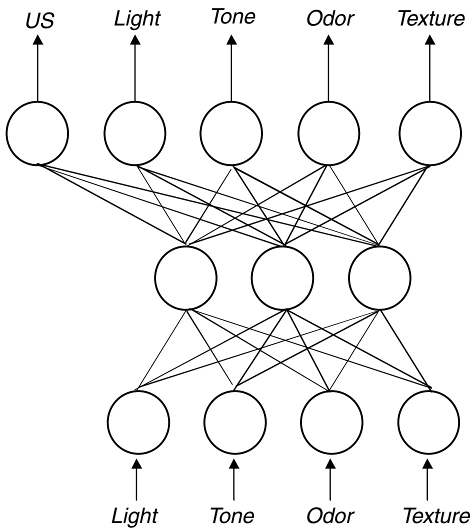


Figure 5.14 A predictive autoencoder is an autoencoder with outputs that reconstruct the inputs *and* predict whether reinforcement (e.g., a US) will occur, given the current inputs. Here, inputs and outputs are labeled to represent four possible sensory stimuli.

For example, suppose that the network is presented with a discrimination paradigm in which one CS (light) predicts the US but the other (tone) does not. The network has one input node corresponding to each possible CS; there are additional input nodes that correspond to the background contextual features (figure 5.15A). For simplicity, the sights, smells, textures, and sounds of the current experimental context are grouped together as context X and represented by a single input node; the sights, smells, textures, and sounds of another experimental chamber might constitute context Y. The network is trained, via error backpropagation, to reconstruct its inputs (whatever CSs and contextual cues are present) and to predict the US on the basis of those inputs.

Initially, presentation of the light in context X will evoke some (random) pattern of activity in the internal-layer nodes (figure 5.15A); presentation of the tone in context X will evoke a different, possibly overlapping representation (figure 5.15B). Presentation of context X alone evokes yet a third representation (figure 5.15C). To master this simple task, the network must adjust the upper-layer weights so that when the internal-layer representation of light is activated, the output nodes corresponding to light, context X, and US are also activated as shown in figure 5.16A. Conversely, when the tone representation is activated, the output nodes for tone and context X—but not

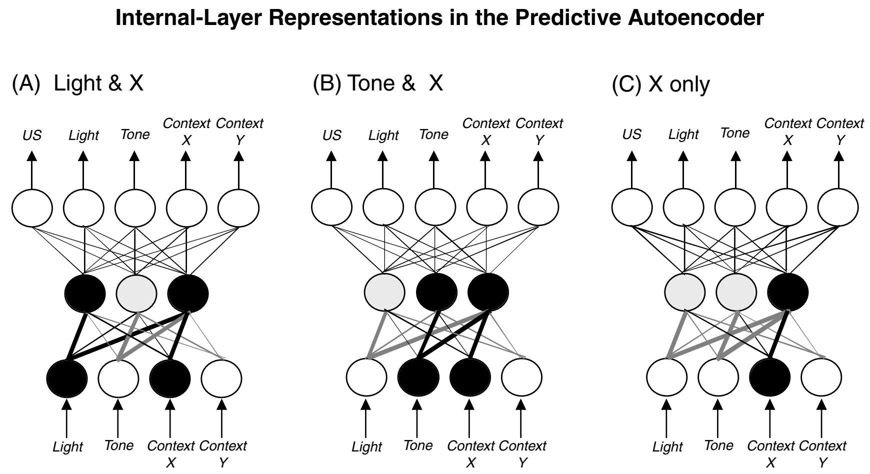


Figure 5.15 Representations in the predictive autoencoder. (A) Presentation of the light in context X activates one pattern of activity in the internal-layer nodes. (B) Presentation of the *tone* in context X activates a second pattern, and (C) presentation of the context X alone—with no explicit CS—generates a third pattern.

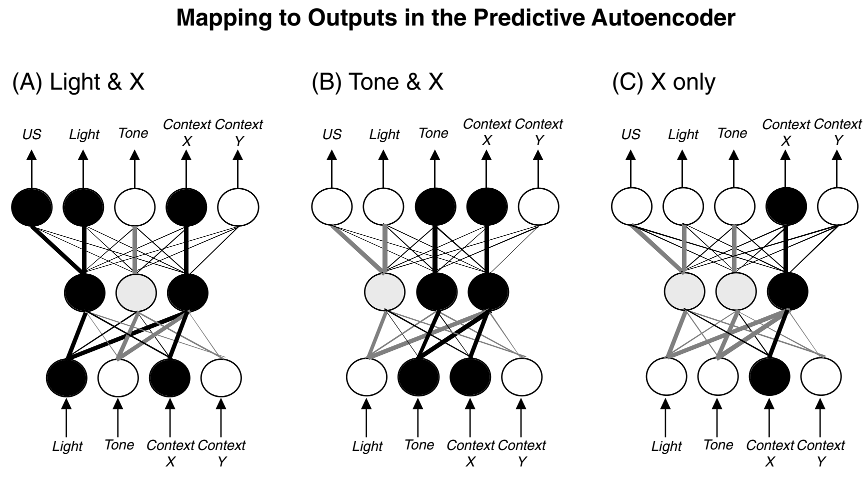


Figure 5.16 The upper layer of weights in the predictive autoencoder learns to map from the internal-layer representations to the correct output. (A) The representation evoked by the light-X input should strongly activate output nodes for light, context X, and the US. (B) The representation evoked by the tone-X input should strongly activate output nodes for tone, context X, and not the US. (C) The representation evoked by the context X alone should strongly activate the output node for X but for no other stimuli.

US—should be activated as seen in figure 5.16B. And, of course, when context X is active but neither light nor tone is present, only the output node for context X should be active, as shown in figure 5.16C.

However, error backpropagation also adjusts the lower-layer weights as well, striving to find a more efficient internal-layer representation. In the case of the current example, the internal-layer representations for light and tone overlap considerably: Both strongly activate the rightmost internal-layer node. Another way of seeing this overlap is shown in figure 5.17A: Light strongly activates the leftmost and rightmost internal-layer nodes, while tone strongly activates the center and rightmost internal-layer nodes. Since light and tone are not redundant (they never co-occur in this example), and

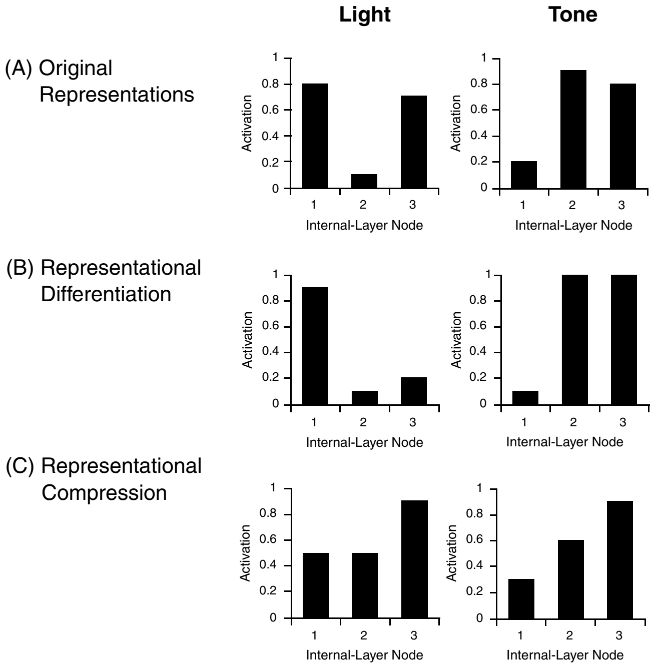


Figure 5.17 Stimulus representations from figure 5.16, encoded as the activity level evoked over a series of nodes. (A) The light stimulus evokes one pattern of internal-layer node activity (strongly activating the leftmost and rightmost nodes), while the tone stimulus activates a different pattern (strongly activating the center and rightmost nodes). (B) If the light and the tone never co-occur, and if they make different predictions about the US, then the autoencoder tends to differentiate their representations, decreasing the overlap. Now, light activates only the leftmost node, and tone activates the other two nodes. In mathematical terms, the difference in representations has increased from *D*(light, tone) = 1.5 in (A) to *D*(light, tone) = 2.5 in (B). (C) Conversely, if the light and the tone co-occur and make similar predictions about subsequent US arrival, their representations are compressed or made more similar. Here, *D*(light, tone) has decreased to only 0.3.

since they make maximally different predictions about the US (light predicts the US, and tone does not), it would be more efficient to **differentiate** their representations. Backpropagation tends to do just this. After many trials pairing light-US and tone-noUS, representations may evolve that are more like those shown in figure 5.17B. Now, there is less overlap, and each internal-layer node is strongly activated by only one cue.

This decreased overlap can be described by subtracting the response of each node to one cue from its own response to the other cues. For example, in figure 5.17A, internal-layer node 1 responds at a rate of 0.8 to light and 0.2 to tone, for a difference of 0.6. The differences in responding to the two cues for the other two nodes are $(0.9 - 0.1) = 0.8$ and $(0.7 - 0.8) = -0.1$. Because we are interested only in the magnitude of the difference, we drop any minus signs, so the differences across the three internal-layer nodes are 0.6, 0.8, and 0.1. Summed across all internal-layer nodes, the difference *D* in representation of light and tone can be calculated as $D(\text{light, tone}) = 0.6 + 0.8 + 0.1 = 1.5$. If the representations are differentiated, as shown in figure 5.17B, *D*(light, tone) rises to 2.5.

While these representational changes are occurring in the lower layer of weights, the upper layer is learning to generate the desired output: reconstructing the inputs and predicting US arrival. Figure 5.18A shows the activation of the network's US-predicting output node on light and tone trials, as a function of training. Initially, responses to both stimuli are low. For about fifty training trials, there is no difference in the response to light and the response to tone. Then the node begins to respond more strongly to light than to tone; by trial 200, the node gives a near-maximal response to light and almost no response to tone. Meanwhile, the internal-layer representation is changing; figure 5.18B shows how *D*(light, tone) slowly increases. Initially, there is little difference between the representations of the two CSs, and for the first fifty trials, the network appears to flounder and *D*(light, tone) actually decreases a bit. At about trial 50, *D*(light, tone) begins to increase. It is at this point that the network has developed representations that distinguish light and tone: Representational differentiation is occurring. Thereafter, the representations continue to differentiate further until the problem is well learned. Note that only after the representations of light and tone begin to be differentiated (around trial 50) does there begin to be differentiation of responses to light and tone in the output layer (compare figure 5.18A).

The predictive autoencoder doesn't only differentiate representations of stimuli that make different predictions about the US; like the standard autoencoder, it also compresses the representations of redundant (co-occurring) stimuli. Suppose that, instead of a light-versus-tone discrimination, the network is trained that light and tone co-occur and both make the same prediction about US arrival (either a prediction that the US will arrive

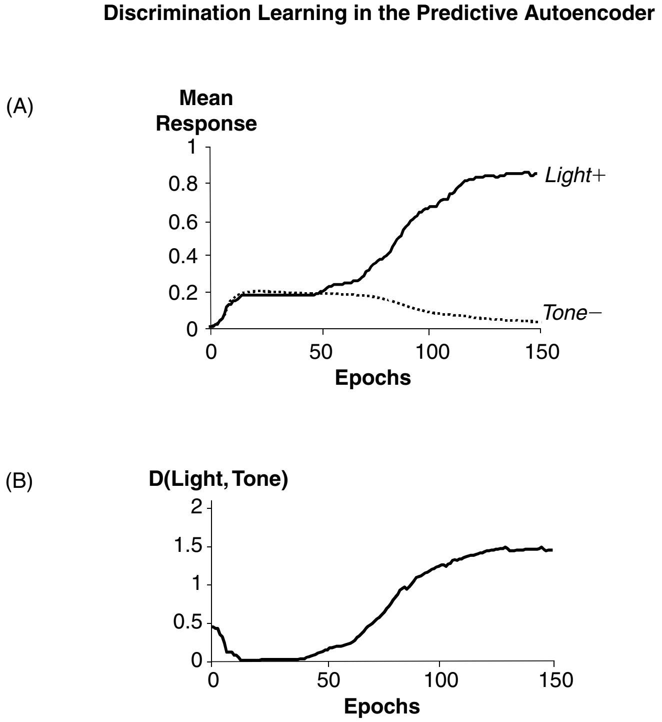


Figure 5.18 Representational differentiation in the predictive autoencoder, trained on light+ and tone−. (A) The activation of the network's US-predicting output node. After about five trials of training, the network begins to predict the US on light+ trials but not on tone− trials. By about trial 150, the responses are strongly differentiated. (B) The differentiation of internal-layer representations, encoded as *D*(light, tone). Early in training, there is little difference in representations. At about trial 50, the internal-layer representations begin to differentiate the two stimuli, and *D*(light, tone) rises. Thereafter, the representations become increasingly differentiated. Note that the differentiation in representations in (B) slightly precedes the differentiated responding in (A).

or a prediction that it will not). In this case, the original representations shown in figure 5.17A become more similar, as shown in figure 5.17C. Each internal-layer node comes to respond similarly to light and tone, and this similarity can be quantified by noting the difference *D*(light, tone) falls from its original value of 1.5 (in figure 5.17A) to only 0.3 (in figure 5.17C).

It is important to note that the degree of compression in a predictive autoencoder does not depend solely on the number of internal-layer nodes. Even with a very large number of internal-layer nodes—enough to devote one internal-layer node to encode every input and output feature—the network still tends to compress the representations of co-occurring inputs (see MathBox 5.1 for further details). This feature of predictive autoencoders has

MathBox 5.1 Hidden Layer Size and Compression

The predictive autoencoder shown in figure 5.16 has an internal layer that contains fewer nodes than either the input layer or the output layer. Thus, if the network is to reconstruct the input, it is forced to compress redundancies in the internal layer, to ensure that nonredundant information makes it through to the output layer. However, even if the number of internal-layer nodes is varied, the same principles of representational compression and differentiation hold.

This is easily shown by example. Suppose that an autoencoder has seven inputs, representing three CSs (A, B, and C) and four contextual cues. The network has eight output nodes, reconstructing the seven inputs and also predicting the US. The network is trained that the stimulus compound AB+ predicts the US but stimulus C− does not.

If the autoencoder has only two internal-layer nodes, then it is necessary to compress the representations of A and B to solve the problem. Then, one internal-layer node can become active when AB is present, and it can activate the output nodes for A, B, and the US; another internal-layer node can become active when C is present and activate the output node for C; when neither internal-layer node is active, only the context output nodes should be active. Thus, as shown in figure A (top), *D*(A, B) is very low—approximately 0—reflecting compression of the representations of A and B, while *D*(A, C) and *D*(B, C) rise, reflecting differentiation of the representation of C from the other two CSs. Within about 2000 training blocks, the network learns to give the correct US prediction to both AB+ and C− (figure A (bottom)).

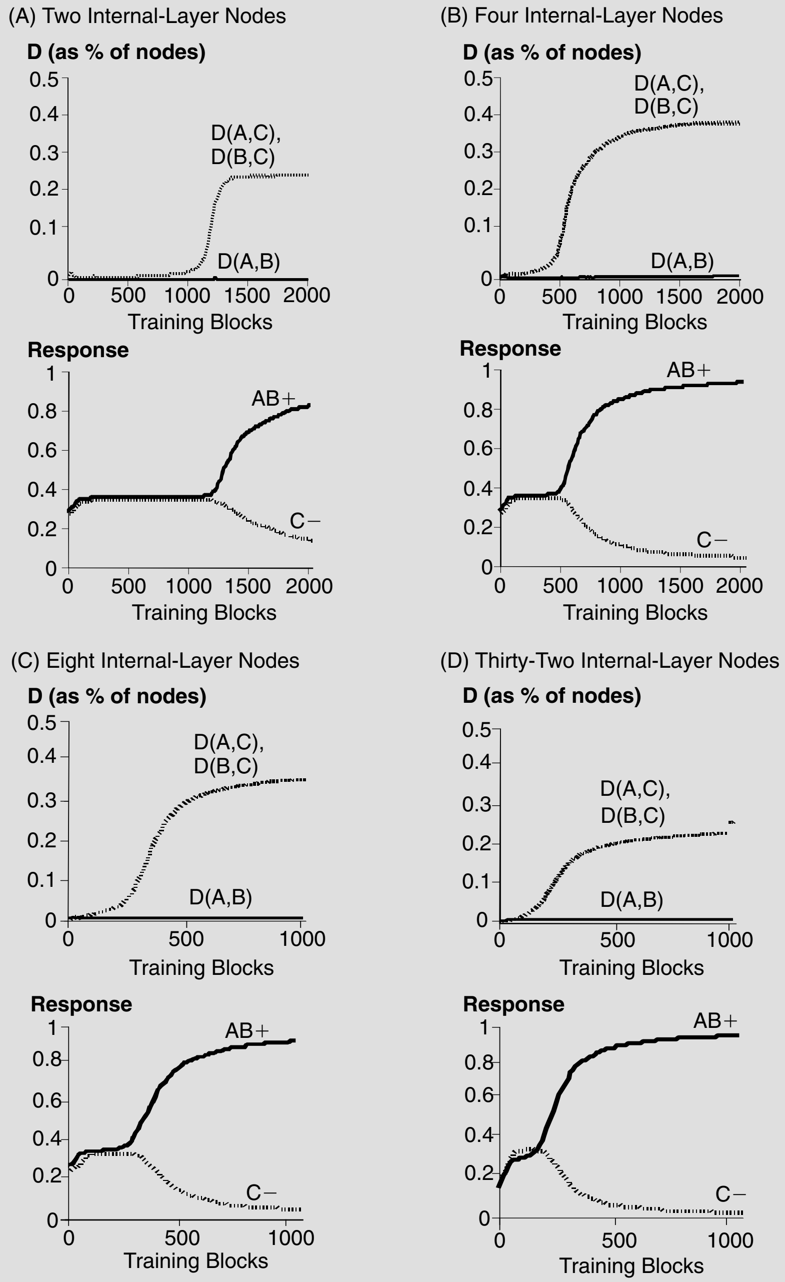
Two internal-layer nodes is the minimum number with which the predictive autoencoder can solve the problem. If the number of internal-layer nodes is doubled, to four, the network can learn the original problem rather more quickly: Only about 1000 training trials are needed to learn AB+ versus C− (figure B (bottom)). However, the pattern of representational changes is largely the same as it was for the two-node case: Representations of A and B are compressed (*D*(A, B) is low) while the representation of C is differentiated from A and B (figure B (top)).

As the number of internal-layer nodes is increased still further, to eight (figure C), there is yet more increase in learning speed, but the basic pattern of representational changes remains the same. Note that with eight internal-layer nodes, it would in principle be possible for the network to adopt a "local" representation, with one internal-layer node coding the presence or absence of each input and one coding the expectation of the US. In this case, once the representations are established, there would be little change in representations even when contingencies are switched: All the same information would be encoded in both phases. However, this is not what happens. The representation of stimuli is generally distributed among all the nodes in the network, independent of how many nodes there are.

If the number of internal-layer nodes is increased still further, to 32 (figure D), more than enough to encode all the information in the inputs, the representations still show the same qualitative pattern, compressing and differentiating representations on the basis of stimulus co-occurrence and meaning.

Thus, as long as there are a certain minimum number of internal-layer nodes, adding additional nodes may facilitate learning but may not cause a qualitative effect in the bias to compress and differentiate representations.

MathBox 5.1 (*Continued*)



(A) Two Internal-Layer Nodes

(B) Four Internal-Layer Nodes

(C) Eight Internal-Layer Nodes

(D) Thirty-Two Internal-Layer Nodes

proven useful in many domains in which a neural network is used to pull out the important statistical regularities in a complex data set. (It is thus similar to, though not identical to, the mathematical technique of principal components analysis.[24])

In summary, *a predictive autoencoder forms new representations biased by two constraints: first, a bias to compress (make more similar) the representations of redundant (co-occurring) stimuli and, second, a bias to differentiate (make less similar) the representations of stimuli that make different predictions about the arrival of the US or other reinforcement*.

In the last several chapters, we have repeatedly used the sensory preconditioning paradigm as an example of how learning occurs from mere expo-

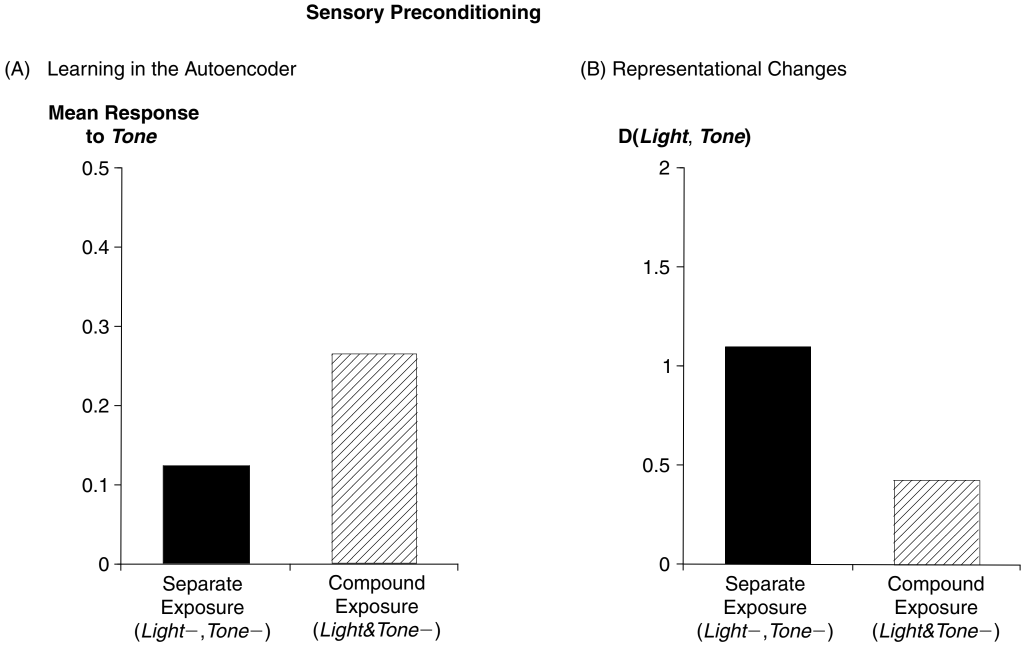


Figure 5.19 Sensory preconditioning in the predictive autoencoder. One group of simulations is given separate exposure to the light and tone stimuli, while the other group is given exposure to the light-and-tone compound. Next, all simulations are given light+ training, pairing the light stimulus and the US. Finally, the tone is presented alone, and the response of the network's US-predicting node is measured. (A) Prior exposure to the light-and-tone compound results in stronger responding to the tone alone than exposure to the light and the tone separately: the sensory preconditioning effect. (B) The representational changes in the autoencoder reflect greater compression (reflected in lower *D*(light, tone)) following compound than separate exposure.

sure to stimulus relationships. To review the paradigm: Unreinforced exposure to a stimulus compound (e.g., light and tone) increases the degree to which subsequent learning about one of the components (light) transfers to the other component (tone).

Animals with hippocampal-region damage do not show sensory preconditioning (refer to figure 2.18); nor does a simple feedforward network such as the ones discussed in chapters 3 and 4. However, a predictive autoencoder can demonstrate sensory preconditioning, as shown in figure 5.19. During phase 1 exposure, the light and the tone are presented together, and neither predicts US arrival; so their representations become compressed. As a result, phase 2 learning about light transfers strongly to tone—more strongly than in a control condition in which there was only separate exposure to the light and tone components in phase 1.

The next chapter will demonstrate that sensory preconditioning is only one of a range of hippocampal-dependent associative learning behaviors that can be demonstrated by a model incorporating a predictive autoencoder.

5.4 INTERIM SUMMARY: WHERE ARE WE NOW?

Chapters 3 through 5 have been largely tutorial, covering the basic principles of error-correction learning, representation, and autoassociation in neural networks. Chapters 6 through 10 will show how these principles can be applied to models of the hippocampus and how it interacts with other brain structures during associative learning.

Before moving on, it seems appropriate to take a few moments to review what has gone before.

Chapter 2 reviewed the kinds of memory impairments that result from hippocampal-region damage in humans and animals. In brief, hippocampal-region damage devastates declarative ("fact") memory but spares many forms of procedural learning. For example, simple CS-US association may be spared, but more complex conditioning involving multiple cues, temporal sensitivity, or latent learning (such as the sensory preconditioning procedure) tend to be disrupted. Electrophysiological studies that record neuronal activity suggest that the hippocampus is intimately involved in all new learning, even the simplest CS-US association, although it may not be necessary for that learning.

Chapter 3 introduced the Rescorla-Wagner model, a powerful model of associative learning from psychology that incorporates an error-correction rule from engineering: the Widrow-Hoff rule. This learning model assumes that there are modifiable weights between nodes representing CSs and USs and that stimuli compete with each other to predict the US. Thus, a CS that predicts the US only some of the time will gain less associative weight as compared to another, co-occurring CS that reliably predicts every occurrence of the US. The error-correction learning embodied in the Rescorla-Wagner model appears compatible with anatomical, neurophysiological, and behavioral data regarding cerebellar circuits. Thus, it is possible that the cerebellum is a circuit-level instantiation of the error-correcting principle of the Rescorla-Wagner model; it is able, on its own, to show just those behaviors accounted for by the model but requiring other brain regions to show more complex behaviors. While the Rescorla-Wagner model can account for many conditioning phenomena, it cannot show configural learning (e.g., negative patterning). It also makes no provision for stimulus representations that encode superficial similarity or allow generalization between stimuli with similar meanings.

Chapter 4 introduced multilayer networks, able to form novel and complex stimulus representations at an internal layer of nodes. One powerful algorithm for training such networks is the error backpropagation algorithm. Multilayer networks can form internal-layer representations that encode arbitrary relationships between stimuli; they can also show configural learning. However, even with this modification, a standard error-correction network fails to show latent learning (e.g., sensory preconditioning). The problem is that error-correction networks learn on the basis of the difference (error) between the US and the network's prediction of the US (measured as the anticipatory conditioned response). In latent learning, in which there is no US, there is no prediction error and hence there is no learning. Interestingly, latent learning is disrupted by hippocampal-region damage.

Chapter 5 began by introducing the concept of an autoassociative network, which learns relationships between co-occurring stimuli. Several features of an autoassociative network—especially high interconnection between nodes—appear in hippocampal subfield CA3. This led many influential modelers to suggest that the hippocampus might operate as an autoassociator. Indeed, an autoassociator is capable of storing arbitrary patterns (memories) and later retrieving them when given partial cues, an ability that seems quite consonant with a hippocampal role in episodic memory. However, autoassociators do not learn explicit CS-US associations, nor are they capable of forming new stimulus representations. The predictive autoencoder is a network that combines some of the most powerful features of all the other networks: It can perform CS-US association and also learn CS-CS associations, and it has an internal node layer, in which new stimulus representations are constructed that compress redundancies while differentiating predictive information—two biases that optimize generalization between stimuli. A predictive autoencoder is capable of latent learning.

Having worked through this material, we are now ready, in the next chapter, to show how such a predictive autoencoder model of the hippocampal region can be used to address a large body of empirical data on the role of the hippocampal-region in associative learning.