3 Association in Neural Networks

This chapter has two basic purposes. First, it is meant to be a relatively painless introduction to neural network models. For this reason, we focus on the simplest possible models and apply them to the most elementary of learning situations: classical Pavlovian conditioning between stimuli (such as tones) and reinforcements (such as airpuffs to the eye). Once such simple models are understood in some detail, it is easier to scale the same principles up to larger models. In fact, many of the more complex models to be discussed in later chapters are simply elaborations of principles that emerge from these smaller systems.

There is a second reason for studying the simplest neural networks: Despite their apparent simplicity, they can provide useful insights into how animals and humans learn, just as the toy airplane models described in chapter 1 can teach us fundamental principles of aerodynamics that apply to full-size planes. Simple network models capture important qualitative ideas about learning and memory expressed within the context of a clearly defined framework for theory development. In chapter 1, we argued that models should be judged not only by how much relevant information they include, but also by how much irrelevant information they leave out. In this chapter, we will introduce one model, the Rescorla-Wagner model, which is in many ways a "model" model. It has stood for nearly thirty years as an example of how it is possible to take a set of complex behaviors, pare away all but the essence, and express the underlying mechanism as an intuitively tractable idea.

Equally important, the Rescorla-Wagner model—for all its apparent simplicity—generates numerous nontrivial predictions that drive empirical studies even to the current day. The Rescorla-Wagner model does not capture all known behaviors, even within its limited domain of classical conditioning.[1] However, even these limitations of the model have proven useful in understanding which types of learning may depend on different processes and thus possibly arise from different brain regions, as discussed later in this chapter. In these ways, the Rescorla-Wagner model represents a case study

for what modeling in the behavioral and neural sciences should aspire to; it is a standard against which the models presented in the rest of this book can be judged.

Section 3.1 reviews the basic properties of nerve cells in the brain and how they communicate with each other. It then introduces the class of models termed **neural networks,** which attempt to capture the essence of this interaction. Section 3.2 discusses how simple (one-unit) models can learn associations between stimuli and reinforcement. Finally, section 3.3 presents the Rescorla-Wagner model of classical conditioning. This model and its variants provide the basis for many of the concepts underlying the more complex models discussed in later chapters.

In this chapter and throughout the book, we present computational principles with a minimum of formalisms; however, some readers may be interested in seeing the formal statements of the models. For these readers, supplemental materials (MathBoxes) are provided with mathematical details and further references. However, the descriptions in the text are sufficient for a reader to understand the material in the rest of the book, without reference to the MathBoxes.

3.1 WHAT IS A NEURAL NETWORK?

Neurons and Information Processing in the Brain

Although the chemical and physical details inside a neuron are exquisitely complex, almost all neurons—in animals and humans—share the same basic architecture and functional principles. Figure 3.1A is a picture of a region of brain tissue with the neurons stained to appear dark. Even within this small region, it is easy to see that neurons vary widely in shape, position, and size. Despite this superficial variation, most neurons have several prototypical features, which are schematized in figure 3.1B. Some of the same basic structure can be seen in the neurons of figure 3.1A.

The prototypical neuron has an **axon,** a primary output pathway by which the neuron sends information to other neurons. The neuron's **dendrites** are treelike branches that collect information sent on by other neurons. In biological jargon, the inputs that arrive at a neuron are called **afferent** inputs (or simply **afferents**).

In most cases, the axon of the afferent neuron comes very close to—but does not touch—the dendrite of the receiving neuron. The gap between neurons is called a **synapse.** Figure 3.2B shows a schematic drawing of a synapse. The afferent neuron releases chemicals, called **neurotransmitters,**

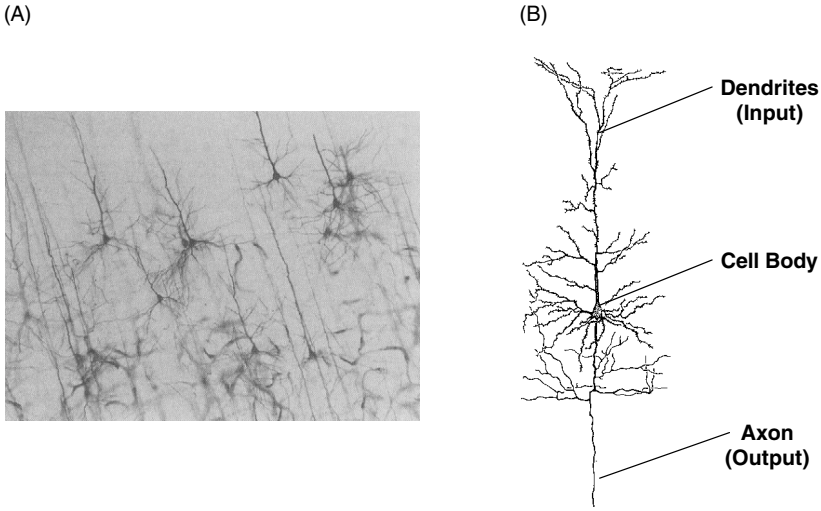


Figure 3.1 (A) Neurons in a small area of brain tissue, revealed through a process known as Golgi staining, in which a random subset of neurons become darkly colored. (Reprinted from Bear, Connors, & Paradiso, 1996, Figure 2.3, p. 25.) (B) A drawing of an individual neuron, in the cortex of a mouse, showing the cell body, dendrites, and axon. The dendrites would be covered with synapses, each a contact from another neuron. The axon is the main output process and may extend some distance before making contact with other neurons. (Adapted from Kuffler, Nicholls, & Martin, 1984.)

into the synapse. The neurotransmitters are picked up by **receptors** in the receiving neuron's dendrites.

The receiving neuron integrates all the information it receives from all its afferents and may produce output in turn, releasing neurotransmitters from its own axon. Some neurotransmitters have an **excitatory** effect on the receiving neuron, meaning that they increase the net activation of the receiving neuron. Other neurotransmitters have an **inhibitory** effect, reducing the net activity of the receiving neuron. Both excitatory and inhibitory inputs can vary in strength, some having a strong effect on the receiving neuron and others having a weaker effect. This depends not only on the particular chemical composition of the neurotransmitters, but also on the strength or efficacy of the synapse itself. Some synapses simply have a greater effect on the overall activity of the receiving cell than others.

Most neuroscientists now believe that a basic mechanism of learning is alteration of synaptic strength. This may occur either by creating new synapses (and deleting defunct ones) or by adjusting the strength of existing synapses.[2]

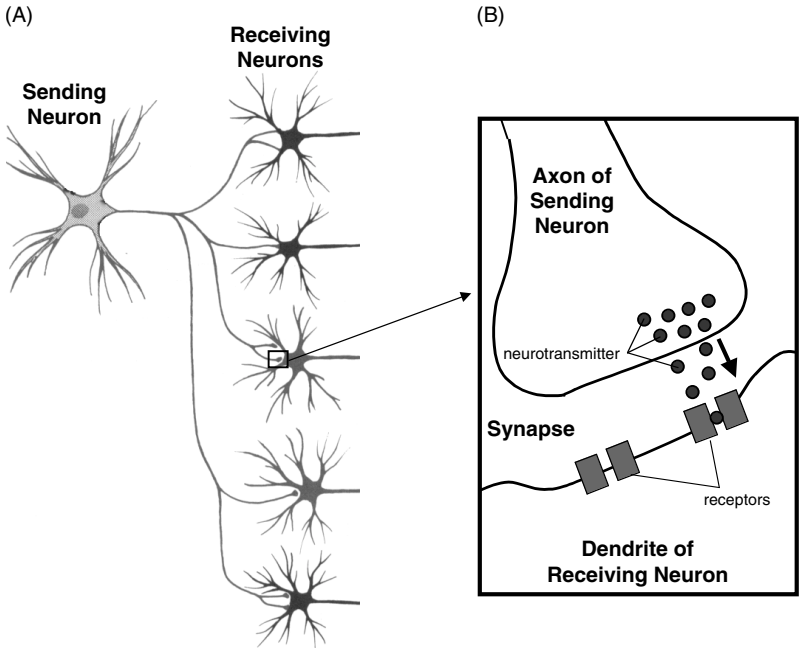


Figure 3.2 (A) An individual neuron sends information via its axon, which may branch to contact many other neurons. The point of contact is called a synapse, a small space between the sending and receiving neuron. (Adapted from Kuffler, Nicholls, & Martin, 1984, Box 1.) (B) Close-up of a synapse. When the sending neuron becomes active, or fires, it releases packets of chemicals, called neurotransmitters, into the synapse. The dendrites of the receiving neuron are covered with receptors, which are specialized to receive different neurotransmitters. The receiving neuron integrates information from all its receptors, and if it receives enough excitatory input, it may become active in turn.

Many subtle details are missing from the preceding description of neuronal processing and communication; some of these will be discussed in later chapters. However, the basic process described above is common to almost all neurons in both animal and human brains. Some of these principles have been known for over a hundred years, and others are still the subject of research. The biggest mystery today is much the same as that pondered by the earliest neuroscientists: How do networks of neurons work together to produce the range of complex behaviors seen in animals and humans?

One approach to understanding how neuronal mechanisms and circuits yield observable behaviors has been to explore simple models of brain circuits. These models allow us to explore the complex behaviors that emerge when neurons are linked together. The study of these model neural networks has gone by many names, including **connectionism** and **parallel distributed**

processing. The study of neural network models has drawn extensively on concepts from neuroscience and psychology, as well as from the tools of mathematics and computer science. Through this interdisciplinary endeavor, modelers have sought an understanding of the general principles by which neuronlike systems can process information and adapt behavior. The next subsection reviews the fundamental principles of modeling neural networks. Later, section 3.2 will discuss the problem of getting networks to learn.

Information Processing in Neural Network Models

Neural network models start with a **node,** a simplified mathematical abstraction of the basic functioning of a neuron. Figure 3.3A shows a network of three nodes, labeled A, B, and C. Each node has inputs (corresponding to a neuron's dendrites) and outputs (corresponding to a neuron's axon). Thus, nodes A and B send output to node *C*, represented by arrows in the figure.

Nodes A and B do not receive inputs from any other nodes in the figure; they are called **input nodes,** because they are the originators of all input to

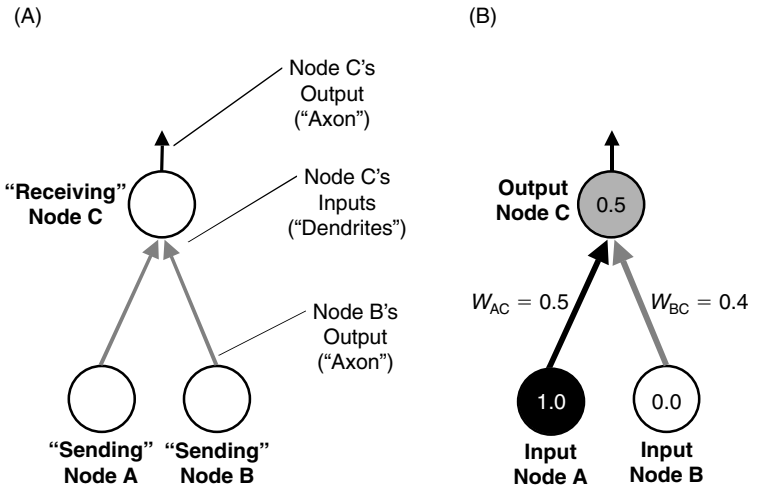


Figure 3.3 (A) Simple model neurons, called nodes. Like neurons, each node has inputs (analogous to a neuron's dendrites) and an output (analogous to a neuron's axon). Here, nodes A and B send output (represented as arrows), which is received and processed by node C. Node C in turn may send output to other nodes (not shown). (B) Each node has an activation level, which determines the probability that the node will fire. These are shown as numbers inside each node. Connections between nodes in a network are often assigned a value, or weight, analogous to the efficacy or strength of a synapse between two neurons. The activation level at node C is determined by the activations of input nodes A and B, multiplied by the connection weights from those nodes. Hence, the activation of node C = $(1.0 \times 0.5) + (0.0 \times 0.4) = 0.5$.

the rest of the network. Node C is called an **output node,** because it does not send output to any other nodes in this network. It may be helpful to think in terms of a rough analogy with the brain: Input nodes might correspond to receptors in the eye or skin that respond to external visual and touch stimuli and send this information on to the rest of the brain. Output nodes might correspond to neurons with axons that leave the brain, such as the nerve cells that guide muscle movement.

Each node in a network has an **activation level,** shown in figure 3.3B as the number inside each node. Activation levels are usually defined to range between 0 and 1. The activation level of a node can be thought of as specifying the probability that the node will generate output of its own. It can also be thought of as approximating the strength of the node's response, on a scale from 0 (not active) to 1 (fully active). In figure 3.3B, we adopt the convention of darkening nodes that are more strongly activated. Thus, input node A is fully active, input node B is not active, and output node C has an intermediate level of activation.

We noted earlier that a combination of excitatory and inhibitory synapses allows one neuron to respond to inputs from a variety of other neurons. Each input can be either excitatory or inhibitory and can be either strong or weak. Within a model neural network, this range of synaptic effects is captured by an **associative weight.** Although most neural network models allow for weights to be both positive (excitatory) and negative (inhibitory), we will begin by confining ourselves to a discussion of positive weights.

Intuitively, associative weights can be thought of as functioning like valves, modulating how much of the activity from one neuron is allowed to flow into the next neuron. If the weights are numbers between 0 and 1, then each weight can be interpreted as a valve that ranges from completely shut (0) to fully open (1), modulating how effective the afferent neuron is in activating the receiving neuron. In figure 3.3B, the weight from node A to C, called W_A, is set to 0.5. This means that 0.5 (50%) of A's activation transfers to C. The weight from node B to C, W_B, is set to 0.4. In the figure, we assume that whatever external stimulus node A responds to is present, so A's activation is 100% or 1.0; B's external stimulus is absent, so B's activation is 0% or 0.0. The activation of *C* depends on the activation of A and B, modulated by the weights. Since $W_A = 0.5$, half of A's activation gets through; since $W_B = 0.4$, slightly less than half of B's activation gets through. The activation at C is the sum of these amounts:

$$\begin{aligned}\text{Activation of C} &= (\text{Activation of A} \times W_A) + (\text{Activation of B} \times W_B)\\ &= (1.0 \times 0.5) + (0.0 \times 0.4)\\ &= 0.5\end{aligned}$$

In this example, the activation of C is the output or product of the network; more complex scenarios are possible, and we will discuss some in the next chapter. If this network were a model of some behavioral system, then the network output could be interpreted as the strength of a behavioral output in response to a particular set of stimulus inputs: stimulus *A* present, stimulus *B* absent.

The output node C in this network performs an information-processing function roughly comparable to that of a neuron: It collects input from a variety of sources, processes that information (in this case, taking a weighted sum), and then possibly generates a response. This might not seem like a very sophisticated information-processing device, especially when compared to the computational capabilities of personal computers, pocket calculators—or the human brain. However, *the power of nodes in a neural network (and neurons in the brain) comes not so much from their individual power, but from the collective power that emerges when many such devices are interconnected in a network.*

Application of Network Models to Motor-Reflex Conditioning

Chapter 2 introduced the experimental procedure of eyeblink conditioning, and we will continue to use this elementary form of learning as an example. To briefly review: Animals are given mildly aversive airpuffs to the eye (the unconditioned stimulus, or US), which naturally evoke protective eyeblinks. If the US is repeatedly preceded by a stimulus cue such as a tone (the conditioned stimulus, or CS), the animal learns a basic association: that the tone CS predicts the imminent arrival of the airpuff US. Eventually, the animal should come to give an anticipatory blink response to the CS cue alone, timed so that the eye is shut when the airpuff US arrives. Learning is assessed by measuring the strength, reliability, and timing of this conditioned response.

Back in 1943, Warren McCulloch and Walter Pitts demonstrated that such associations could be modeled using a simple neural network much like the one shown in figure 3.4A.[3] A possible CS, such as a tone or light, is assigned to each input node; the activation level of an input node is set to 1.0 when the corresponding cue is present and to 0.0 otherwise. The activation of the output node is interpreted as the strength or probability of a conditioned response, such as an eyeblink. In the example shown in figure 3.4A, the tone is present and the light is absent. Before any training, we assume that all weights between nodes are set to 0.0. Thus, the presentation of the tone does not initially evoke any activation in the output node, and there is no conditioned response.

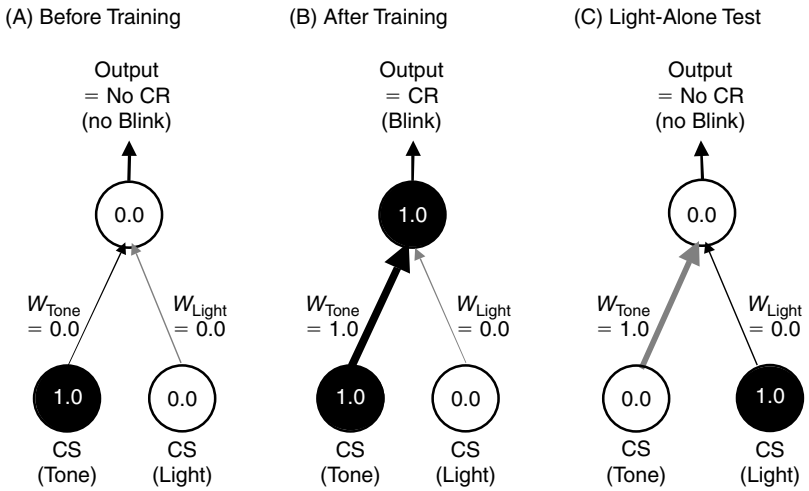


Figure 3.4 (A) A one-layer network model of eyeblink conditioning with input nodes, representing tone and light CSs, and one output node. The activation of the output node is the network's response and corresponds to the strength or probability of a CR (eyeblink). Weights are initially set to 0.0, so activation in either CS node does not produce any activation in the output node nor any CR. (B) The status of the network after being trained that the tone predicts the airpuff. Now the weight from the tone CS has been strengthened to 1.0 (represented by the thicker arrow), and so activation of the tone CS node produces activation in the output node and a blink CR. (C) Since the weight from the light CS has not been changed, activation of the light CS node still does not produce a CR.

Figure 3.4B illustrates what the same network might look like if it had been trained to give a response to the tone CS but not the light CS. In this figure, the weight from the tone node is 1.0 (schematized by a thicker arrow in the figure). Now, whenever that CS is present, the activation of the output node will be $1.0 \times 1.0 = 1.0$; thus, the network generates a strong response to the tone CS. On the other hand, the light CS still has a weight of 0.0; thus, when the light is present (figure 3.4C), the activation of the output node will be $1.0 \times 0.0 = 0.0$, and the network will not respond.

Note that in this very simple model, there are no assumptions about how sensory stimuli such as lights and tones are preprocessed nor how conditioned responses are translated into motor actions. These functions are important, but they take place outside the scope of this model. Like airplane propulsion for the engineer studying aerodynamics in chapter 1, these issues are not directly relevant to the memory processes being studied—and indeed might encumber the model with unnecessary additional complexity, detracting from the model's explanatory power and focus. The model in

figure 3.4 is meant to address only one issue: How does one CS come to evoke a response but another does not? The model suggests that this process can be understood by means of associative weights that influence responding.

A reasonable question to ask now is: How does one train the naive network in figure 3.4A so that it becomes the fully trained network of figure 3.4B? The only difference between these two networks is in their weights. *Thus, the key to learning in neural networks is changing the weights between nodes.* As we noted earlier, this is intended to roughly capture what happens during learning in networks of real neurons, in which synaptic strengths are altered, thereby altering the ability of some neurons to cause firing in other neurons.

The earliest approaches to the problem of how to change the weights in a neural network came neither from psychology nor from neuroscience, but rather from engineering. The next section presents an early engineering approach to learning in neural networks.

3.2 NEURAL NETWORK MODELS OF LEARNING

In the late 1950s and early 1960s, Bernard Widrow and Ted Hoff were using neural networks to try to solve some classic problems in engineering and signal processing. These included the automatic transcription of speech, recognizing objects from novel angles, and diagnosing heart defects on the basis of EKG recordings. People can be trained to do all these tasks very well, but it is hard to codify the rules into a traditional computer program. Neural networks offered an alternative approach: Instead of having a programmer tell the computer how to perform a task, the computer could learn for itself.

Widrow and Hoff set up their engineering problems so that each could be represented as a set of input patterns, each of which was associated with a desired output pattern. For example, in speech-to-text transcription, the first step was to preprocess the speech into a set of patterns that could be applied as input to a network. This kind of preprocessing is a complex process and the subject of much research.[4] These input patterns were then paired with output patterns that represented instructions to generate typewritten text. The neural network's job was to take each input pattern and generate the desired output, in this way translating "speech" into "text."

The key problem for Widrow and Hoff was how to teach a network this kind of mapping from input to output. For a simple problem such as eyeblink conditioning, it is often possible to intuit what the fully trained network should look like. For example, in figure 3.4, setting values of 1.0 and 0.0

on the tone weight and the light weight, respectively, made the network respond to the tone but not to the light. In real brains, of course, synaptic strengths are not set "by hand" but learned through repeated exposures to regularities in the environment. Moreover, in very complex problems such as speech transcription, the network might need hundreds or thousands of weights to encode highly complex relationships between the input patterns and the output patterns. In such a case, it is difficult or impossible to choose all the appropriate weights by hand. Thus, Widrow and Hoff needed a **learning rule:** an algorithm that would allow the network to adapt its weights to solve an arbitrary problem.

The Widrow-Hoff Learning Rule

Widrow and Hoff developed a simple but effective learning rule known today as the Widrow-Hoff rule.[5] The rule also goes by several other names, including the LMS (least-mean-squared) or delta rule, for reasons relating to its mathematical underpinnings. These details are presented in MathBox 3.1 for those who wish a deeper understanding of these formal underpinnings; however, they are not essential to understanding the rest of the material in this book. The Widrow-Hoff rule and other related algorithms[6] formed the foundation for tremendous subsequent growth of neural network models in engineering, neuroscience, and psychology.

To train a network to predict or classify inputs correctly, Widrow and Hoff noted that there needs to be an external signal, distinct from the network's actual response, that specifies the desired output. This signal, often called a **teaching input,** is shown in figure 3.5. In the context of eyeblink conditioning, the teaching input specifies whether the airpuff US occurred and thus whether the **desired output** was 1.0 (if the airpuff US was present) or 0.0 (if the airpuff US was absent). Given such a teaching input, the network can compare its own output against the desired output and then adjust the weights so that, in the future, errors are less likely.

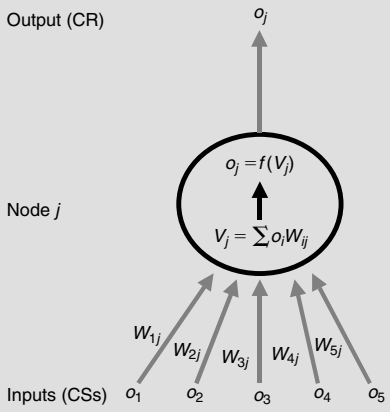
Widrow and Hoff proposed that this process should work as follows: First, cues are presented; input nodes corresponding to those cues become fully active (activation level = 1.0). This activation propagates through weighted connections to the output node, producing some level of activation there. This activation becomes the node's output and indicates the strength or probability of a conditioned response.

Next, the teaching signal provides information about desired output. In the case of eyeblink conditioning, this desired output, as shown in figure 3.5, is 1.0 if the airpuff US occurred and 0.0 if the airpuff US did not occur.

MathBox 3.1 The Widrow-Hoff Rule

The Widrow-Hoff rule is an algorithm for training a simple processor to learn arbitrary mappings between input patterns and output patterns.

The basic unit, or *node,* is shown below. Each node j receives a series of inputs i, each of which has a value o_i. In the simplest case, $o_i = 1.0$ if input i is present and 0.0 otherwise, although intermediate values may be assumed. There may be multiple nodes, all operating in parallel and each learning a different relationship from inputs to outputs.



Each node processes its inputs according to an *activation rule.* The total activation V_j of node j is defined as the weighted sum of all inputs i present on the current trial:

$$V_j = \sum_i o_i w_{ij} \tag{3.1}$$

In this rule, w_{ij} is the *weight* or strength of the connection from i to j. If $w_{ij} > 0$, then when input i is present, it increases the total activation of j. If $w_{ij} < 0$, then when input i is present, it inhibits the total activation of j. If $w_{ij} = 0$, then the presence or absence of input i has no effect on j. (Note: In this and other boxes, the terminology has sometimes been changed from the original publication to maintain notational consistency throughout the text.)

Next, an *output rule* is used to convert activation to node output:

$$o_j = f(V_j) \tag{3.2}$$

In the simplest case, the output function f is merely an identity function ($f(x) = x$), so Equations 3.1 and 3.2 combine to yield

$$o_j = \sum_i o_i w_{ij} \tag{3.3}$$

This means that the output of node j is simply the sum of the weighted activations of all inputs i present on the current trial.

Next, the *error* ∂_j for node j is computed as the difference between the node's desired output d_j and its actual output o_j:

$$\partial_j = (d_j - o_j) \tag{3.4}$$

Finally, the weights are changed to reduce this error. For every weight w_{ij} from an input i to node j,

$$\Delta w_{ij} = \beta \delta_j o_i \tag{3.5}$$

Δw_{ij} is the amount by which w_{ij} changes on the current trial. If $\Delta w_{ij} > 0$, then the weight increases; if $\Delta w_{ij} < 0$, then the weight decreases. β is the network's *learning rate*. This is a fixed small number that determines how much a weight may be changed on a single trial. Typically, β might be set to a value between 0.01 and 0.1; in the example in Section 3.2, $\beta = 0.2$ for ease of explanation.

Mathematically, it can be shown that repeated applications of the Widrow-Hoff rule, with a sufficiently small value of β, will improve performance by minimizing the squared difference between actual and desired outputs across all input patterns; for this reason, the Widrow-Hoff rule is also known as the LMS (least-mean-squared) rule. It is also sometimes called the delta rule.

The reader who is interested in further mathematical details may refer to McCulloch & Pitts (1943); Rosenblatt (1958); Widrow & Hoff (1960); Rumelhart, Hinton, & Williams (1986); Gluck & Bower (1988a); Widrow & Winter (1988); Minsky & Papert (1998).

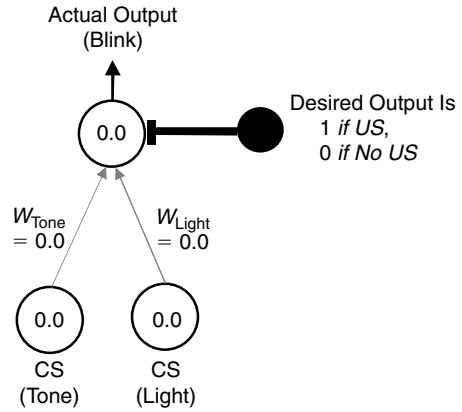


Figure 3.5 A one-layer network for eyeblink conditioning showing the addition of a teaching node indicating the desired output, as suggested by Widrow and Hoff (1960).

Widrow and Hoff defined the network's **error** on a particular trial as the difference between the desired output and the actual output, namely,

OutputError = (DesiredOutput − ActualOutput)

For example, consider the trained network of figure 3.4B. The tone CS is presented, and the output node produces a response (ActualOutput = 1.0). The US will arrive on that trial, so the DesiredOutput = 1.0. Therefore, we can calculate the OutputError as follows:

$$\begin{aligned}\text{OutputError} &= (\text{DesiredOutput} - \text{ActualOutput}) \\ &= 1.0 - 1.0 \\ &= 0.0\end{aligned}$$

If the same network is tested on a trial in which the tone CS is absent (e.g., figure 3.4C), the network will not respond (ActualResponse = 0.0), and the US will not occur (DesiredResponse = 0.0). Hence, the Output Error is calculated as

$$\begin{aligned}\text{OutputError} &= (\text{DesiredOutput} - \text{ActualOutput}) \\ &= 0.0 - 0.0 \\ &= 0.0\end{aligned}$$

In both these cases, the network's response is perfectly correct, there is no error, and we can conclude that for these trials, the network's weights are as they should be.

However, under other circumstances, there can be an error. For example, consider the untrained network of figure 3.4A: The tone cue is presented, the network does not respond, but the US arrives:

$$\begin{aligned}\text{Output Error} &= (\text{DesiredOutput} - \text{ActualOutput}) \\ &= 1.0 - 0.0 \\ &= 1.0\end{aligned}$$

As the network is trained with repeated trials, the error should gradually decrease toward 0.0 (figure 3.4B). Thus, the difference between an untrained network (figure 3.4A) and a trained one (figure 3.4B) is all in the weights, as was noted earlier. Learning is a process of slowly changing the weights to produce appropriate responding—and reduce the error toward 0.0.

In this kind of neural network, the blame for an incorrect response (OutputError > 0) or the credit for a correct response (OutputError $= 0$) lies in the weights. But there may be many weights in the network, and not all of them may deserve the blame or credit equally. This is a fundamental problem for training neural networks and is often referred to as a credit assignment problem because the issue is how to assign credit (or blame) for a network's output performance to the weights in the network.

Widrow and Hoff's rule provides one solution to the problem of credit assignment. Their approach is called *error-correction learning* because it specifies that each weight should be adjusted according to its particular contribution to the total network error.

The first issue is determining which weights to adjust. Recall that the rule for computing output response is a weighted sum: Each input node contributes to the output node's activation as a function of its own activation multiplied by its weight. If an input node's activation is 0, then it has made no contribution to the output node. Only input nodes with activation > 0 can have any influence.

Thus, if there was some error and the output node's response was wrong, the fault must lie among the weights of those input nodes that were active on the current trial. Under these circumstances, the Widrow-Hoff rule will "punish" or reduce all these weights from the active nodes. This might seem like a rather heavy-handed approach, roughly equivalent to the police arresting all those who were present at the scene of the crime. But each weight is changed by only a very small amount on each trial. A particular weight that is active on one trial when the node's response is wrong will be changed only slightly; but if that same weight is active repeatedly, on many trials when the response is wrong, these changes will accumulate. Over time,

the weights that are most to blame for the error will be changed the most; and given enough trials, the network will find an optimal set of weights that minimizes the total error over all types of trials.

To better understand how this learning rule operates, consider the example shown in figure 3.4A, in which the network fails to respond on a trial in which the tone cue is present and the resulting output response is incorrect (OutputError = 1.0). The fault must lie in the weight from this active tone node but not in the weight from the inactive light node. If there had been other cues present, then the weights from those input nodes would have been changed too.

Now that the blame has been assigned to a subset of the network's weights, the second issue is how much to change each weight. A parameter known as the **learning rate** determines how large the weight changes should be. If there is a large learning rate, then the network as a whole will quickly adapt itself to new contingencies. This can be good or bad; if the learning rate is very high, then random fluctuations in the world (such as noise on the inputs) can lead to large changes in the network, even to the point of disrupting previously stored knowledge. On the other hand, a very low learning rate means that the network will be very stable but will take a long time to adapt to new contingencies. This paradox, called the plasticity-stability trade-off, has been the focus of much study.[7] There is no fixed rule for how to set the learning rate so as to achieve the best balance between stability and plasticity; the optimal learning rate depends on the particular problem being learned. We will return to the topic of learning rates in chapter 10 when we discuss possible neurobiological mechanisms for adjusting learning rates in the brain.

One way to improve a network's performance (that is, to minimize the likelihood of errors) is to make the weight changes depend not only on a fixed learning rate, but also on the current error. If the error is high, then there is good cause to make large changes in the network—even if some prior learning is disrupted as a result. On the other hand, if the error is small, then the network is performing relatively well, and changes should be smaller to preserve the prior learning.

Widrow and Hoff argued that weight changes should depend on both a fixed learning rate and the current error. Thus, for all nodes active on a given trial, weights are corrected for the next trial according to the following:

$$\text{NewWeight} = \text{OldWeight} + (\text{LearningRate} \times \text{Error})$$

An example may help to clarify how this works. Figure 3.6A is a copy of the untrained network of figure 3.4A. Weights from both input nodes are initially set to 0.0. On the first trial, the tone is presented. The output node's activation is the sum of the weighted activation of each input: 1.0×0.0 from the tone

and 0.0×0.0 from the light. The activation is therefore 0.0, and the output is also 0.0. Since the airpuff US was present on this trial,

$$\begin{aligned}\text{OutputError} &= (\text{DesiredOutput} - \text{ActualOutput}) \\ &= 1.0 - 0.0 \\ &= 1.0\end{aligned}$$

Now, according to the Widrow-Hoff rule, this error should be distributed among active input nodes—only the tone in this case—as follows:

$$W_{\text{Tone}} = 0 + (\text{LearningRate} \times 1.0)$$

Suppose the learning rate is defined as 0.2; then the result of the first tone-airpuff pairing should be

$$W_{\text{Tone}} = 0 + (0.2 \times 1) = 0.2$$

Figure 3.6B shows the updated weight. The network has just taken its first step toward learning the correct response.

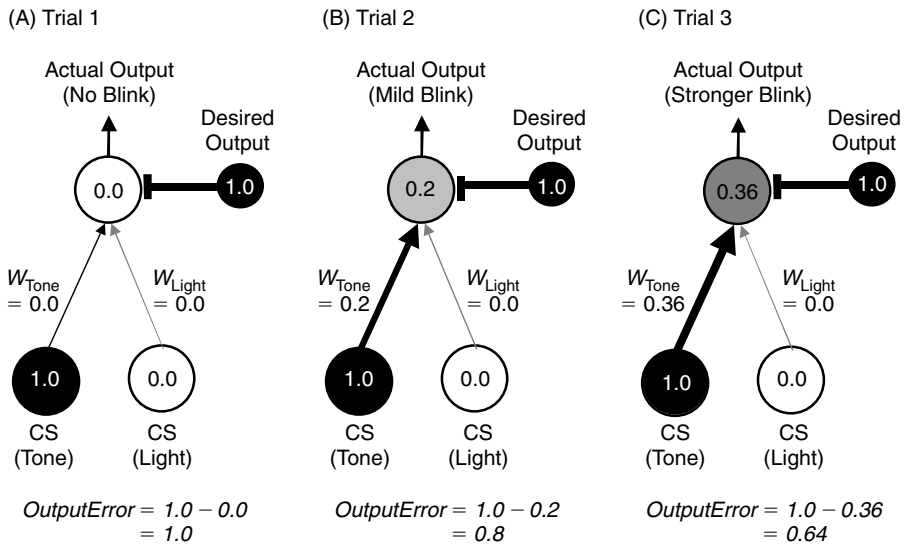


Figure 3.6 Learning to respond to the tone CS. (A) The untrained network before any training. The tone CS is presented, but since its weight is 0.0, there is no activation of the output node, and there is no response. Since the airpuff US always follows the tone CS, the desired output was 1.0, so the output error is 1.0. (B) Assuming a learning rate of 0.2, the weight from the tone CS is adapted to 0.2. On trial 2, the tone CS is presented again. This time, the actual output is 0.2, so the output error is reduced to 0.8. (C) Again, the weight is updated, and on trial 3, the actual output is 0.36, and the output error is only 0.64.

On the second trial (figure 3.6B), the tone occurs again. But this time, weight from the tone is 0.2, so the output node's activation is $1.0 \times 0.2 = 0.2$. This means that the output node is giving a weak response, at 20% of maximum strength (or, alternatively, has a 20% probability of firing). This is clearly better performance than on the last trial, although there is still room for improvement. On the next trial the error is calculated as

$$\begin{aligned}\text{OutputError} &= (\text{DesiredOutput} - \text{ActualOutput}) \\ &= 1.0 - 0.2 \\ &= 0.8\end{aligned}$$

Once again, the Widrow-Hoff rule specifies that the weight from the active input node should be adjusted as

$$W_{\text{Tone}} = 0.2 + (0.2 \times 0.8) = 0.36$$

On the third trial (figure 3.6C), the output is $1.0 \times 0.36 = 0.36$. Again, the response is better than before; the network is improving, slowly, with each trial. If the process is repeated, with more and more pairings of the tone and the airpuff US, the output response will gradually increase toward 1.0 while the output error will gradually decrease toward 0.0. The error is also decreasing ($1.0 - 0.36 = 0.64$). Figure 3.7A shows how the output response increases across 15 repeated trials, and figure 3.7B shows how the output error decreases during these same 15 trials.

Figure 3.7A is a **learning curve,** and it is directly comparable to the kind of learning curve generated in an experiment in which animals (or people) are trained to give a conditioned response (e.g., figure 3.7C,D). The most important characteristic is the curve's shape: Early in the acquisition phase, there are large changes in the response between trials, and the network output shows rapid improvement. Later in the acquisition phase, the response levels off near 1.0, and there are only small changes in response from trial to trial. This leveling-off point is called the **asymptote.**

Although all learning curves are roughly the same shape, the labeling and calibration of the axes may be very different. Figure 3.7C shows data from rabbits that are learning conditioned eyeblink responses. On average, the rabbits take some 300 trials to acquire the response. By contrast, figure 3.7D shows data from dogs that are learning that a light predicts arrival of a food US when the conditioned response measured is the degree of anticipatory salivation. Dogs learn the response in about twelve trials. These two kinds of learning occur at different speeds, but the overall shape of the acquisition curves are remarkably similar. Similarly, in the model data (figure 3.7A), the speed of learning can be adjusted upward or downward to change the speed

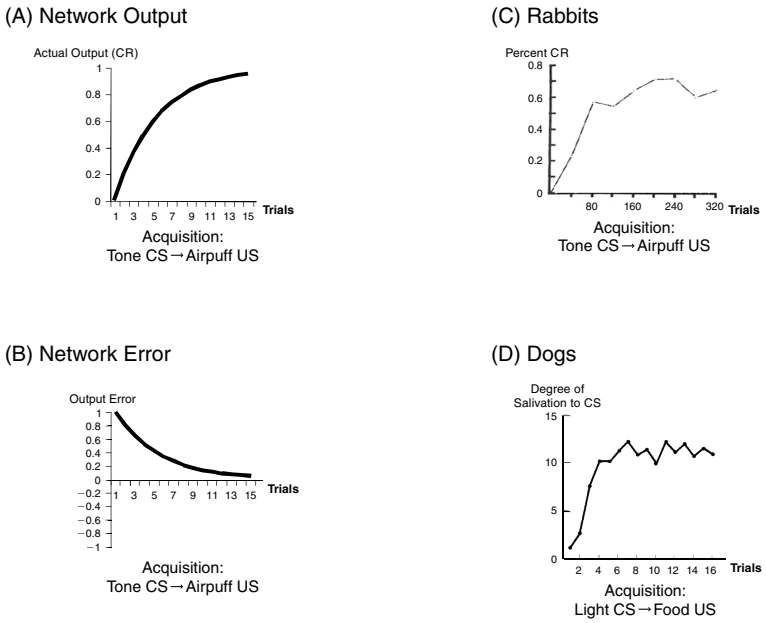


Figure 3.7 (A) Learning curve for the network shown in figure 3.6, given 15 acquisition trials pairing CS and US. Initially (trial 0), the network does not respond to the CS at all, but response increases with repeated pairings; the strength of the response (CR) climbs quickly over the first few trials and then levels off near 1.0. This leveling-off point is the asymptote. (B) The output error corresponding to the learning shown in A. Early in acquisition training, when the actual response is low, the error is high; as the actual response grows, error drops off to zero. (C–D) The learning curves for CS-US acquisition training in rabbits (C) and dogs (D) are similar to the model. (C is adapted from Gormezano, Kehoe, & Marshall, 1983; D is adapted from Hilgard, Atkinson, & Atkinson, 1975, Figure 7.4, p. 197.)

of learning. The actual number of trials required to acquire a response in the model is less important than the fact that the overall shape of the learning curve should be similar to that seen in animal data.

In animals, if the **acquisition** trials (CS-US pairings) are followed by **extinction** trials (CS alone, with no US), the initial learned response will gradually disappear (or *extinguish*). Figure 3.8C,D shows what happens to the learning curves of figure 3.7 when animals are given such additional extinction trials. Similarly, if the network is given CS-US pairings followed by CS-alone extinction trials, it shows similar behavior (figure 3.8A). On the first extinction trial, there is no US, and so the desired output is 0.0. However, because of the previous training trials, the actual output is 0.96 (close to 1.0); the error is computed as follows:

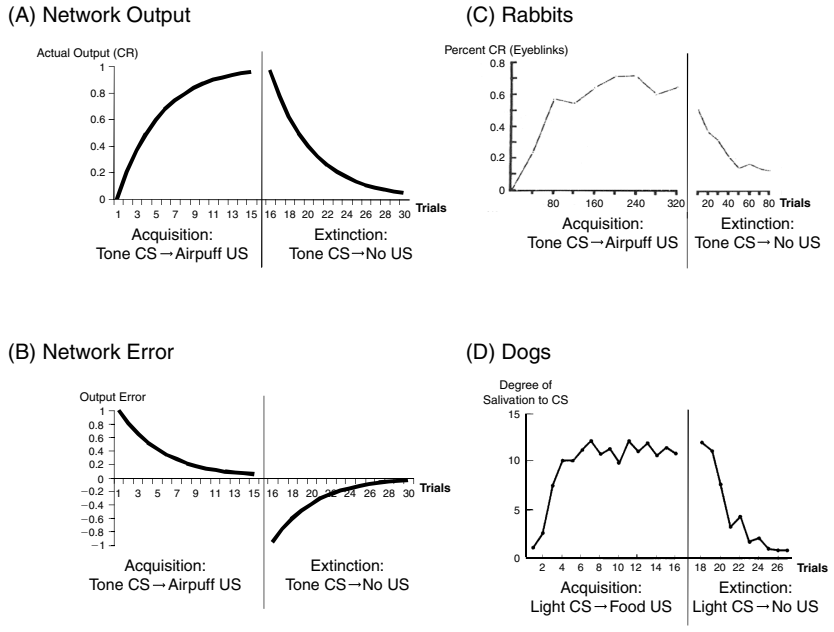


Figure 3.8 (A) After 15 trials of CS-US learning, the network from figure 3.6 is given 15 extinction trials, in which the CS is presented without the US. Again, the response changes quickly over the first few extinction trials, dropping from its previously high levels, then leveling out at a new asymptote near 0.0. (B) The output error is large early in the extinction phase, while the network is still responding to the CS but no US arrives. As the network response in A decreases, the error returns to 0.0. (C–D) Extinction in animals (C = rabbits, D = dogs) shows the same basic form as the model in A. (B is adapted from Gormezano, Kehoe, & Marshall, 1983, Figure 3B; D is adapted from Hilgard, Atkinson, & Atkinson, 1975, Figure 7.4, p. 197.)

$$\begin{aligned}\text{OutputError} &= (\text{DesiredOutput} - \text{ActualOutput}) \\ &= (0 - 0.96) \\ &= -0.96\end{aligned}$$

Note that now the error is a negative number, indicating that the actual output activation was too high. Intuitively, we might therefore expect that the weight from the active nodes should be reduced, and this is precisely what the Widrow-Hoff rule does:

$$\begin{aligned}\text{New}W_{\text{Tone}} &= \text{Old}W_{\text{Tone}} + (\text{LearningRate} \times \text{Error}) \\ &= 0.96 + (0.2 \times -0.96) \\ &= 0.77\end{aligned}$$

The new weight (0.77) is noticeably smaller than the old weight (0.96). Thus, on the second extinction trial, the network's response to the tone CS will be a little

weaker. With repeated extinction trials, the response fades away altogether, as is shown in the right half of figure 3.8A. Thus, the Widrow-Hoff error-correcting learning procedure works to adapt the network to the regularities in the environment, even when these regularities are altered. Whereas on trials 1 through 15, the US reliably followed the tone, this has now changed, and the US no longer follows the tone on trials 16 through 30. Tracking these changes in environmental regularities, the network adapts its weights so as to continue to seek an optimal set of weights for correctly predicting the US.

The actual Widrow-Hoff rule was slightly more complex than the foregoing discussion implies. It allowed for the input nodes to be partially active and for a few other complications. Full mathematical details are given in MathBox 3.1.

Widrow and Hoff's work in the late 1950s and early 1960s led to a wide range of computing devices used by engineers for tasks as diverse as weather prediction and the mechanical control of balance.[8] Widrow himself continued to refine the approach and produced networks that could learn difficult problems including recognizing rotated patterns, performing time-series prediction, playing the game of blackjack, and backing a tractor-trailer into a loading dock.[9] Widrow's simple networks also found many practical uses in signal processing applications, including the reduction of noise in computer modems, long-distance telephone lines, and satellite communications.[10] Widrow's collaborator Hoff did not continue to work in this area; he and a handful of fellow Stanford students left academia to start a computer company. They called it Intel. But that's another story.

3.3 RELATIONSHIP TO ANIMAL LEARNING

Widrow and Hoff were engineers, studying machine learning because they wanted to create intelligent computers. They weren't particularly concerned with whether their learning algorithms bore any meaningful resemblance to learning in the brain—any more than an engineer designing airplanes might care whether the designs capture any features of bird flight.

However, a decade after Widrow and Hoff developed their neural network learning rule, psychologists realized that some very fundamental kinds of animal learning could also be described by the same basic error-correcting principle. This insight led to the development of the Rescorla-Wagner model,[11] a simple but elegant description of classical conditioning in animals and humans. As described below, this new model, and the experiments that led to it, challenged some basic axioms of learning that dated back over seventy years.

The Blocking Effect

At the turn of the century, Ivan Pavlov had argued that classical conditioning resulted solely from the pairing of a CS, such as a tone or bell, and a US, such as the presentation of food.[12] That is, as long as the CS and US occurred closely together in time, association would develop between them. By the late 1960s, however, several studies had shown that conditioning wasn't quite so simplistic: It wasn't enough for CS and US to merely co-occur; instead, they had to have some meaningful, predictive relationship.

A classic experiment that demonstrates this principle is the **blocking** paradigm, introduced by Leo Kamin in 1968, which follows the procedure outlined in table 3.1. Kamin originally demonstrated blocking using rats that had been trained to press a lever for food,[13] but the same effect can be obtained in a variety of preparations, including rabbit eyeblink conditioning.[14]

First, the animals are divided into two groups, called the Sit Exposure group and the Pre-Trained group. In phase 1, each animal in the Pre-Trained group is trained that a tone CS predicts an airpuff US. Training continues until the animal learns to give a reliable eyeblink response to the tone. In contrast, the animals in the Sit-Exposure group sit in the experimental chamber for an equivalent amount of time, but they are given no exposure to the tone or the airpuff US. Next, in phase 2, each animal (in both groups) is given presentations of a compound stimulus, consisting of the old tone CS plus a new light CS, presented simultaneously; this compound tone and light stimulus is always followed by the airpuff US. Animals in the Pre-Trained group, having already learned to respond to the tone, continue to respond to the tone and light compound. Meanwhile, animals in the Sit-Exposure group also learn to respond to the tone and light compound. Finally, in phase 3, all animals are tested for responding to the tone alone and to the light alone. Figure 3.9 shows a typical set of results from a blocking experiment in rabbits.[15] Animals in the Sit-Exposure group give reliable responses to either CS.

Table 3.1 The Blocking Experiment

Group	Phase 1	Phase 2	Phase 3: Test
Pre-Trained	Tone CS → airpuff US	Tone CS and light CS → airpuff US	Tone CS → ? Light CS → ?
Sit-Exposure	(*Animal sits in experimental chamber*)	Tone CS and light CS → airpuff US	Tone CS → ? Light CS → ?

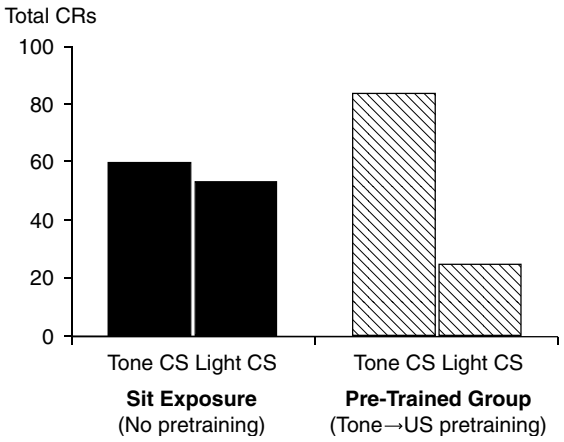


Figure 3.9 Phase 3 responding for a blocking experiment like that outlined in table 3.1. Rabbits in the Sit-Exposure group show strong responding to both the tone CS and the light CS. Rabbits in the Pre-Trained group show strong responding to the tone CS but very little responding to the light CS. Prior training to the tone CS in phase 1 blocks subsequent learning about the light CS in phase 2. (Plotted from data presented in Solomon, 1977.)

By contrast, animals in the Pre-Trained group respond strongly to the tone but give virtually no responses to the light. It appears as if prior learning about the tone blocks subsequent learning about the light.

This blocking effect was problematic for early theories of conditioning, because they expected learning to be determined only by the co-occurrence of the CS and US. If this were true, animals that were given light and US pairings in phase 2 should have learned an association between the light and the US, regardless of prior training. Yet this was not the case: Animals in the Pre-Trained group, which had previously learned to respond to the tone, showed very little responding to the light in phase 3.

Apparently, co-occurrence alone is not sufficient for learning. Instead, blocking suggests that *learning occurs only when the CS is a useful predictor of the US*.[16] In the Pre-Trained group, the tone is a perfectly reliable predictor of the US in phase 1. When phase 2 begins, the tone continues to predict the US; the presence of the light adds no useful additional information. Therefore, there is little learning about the light in phase 2, resulting in weak responding to the light in phase 3.

Blocking also occurs in human learning. For example, in a study by Tom Trabasso and Gordon Bower, college students were trained to categorize objects according to certain predefined rules.[17] The students were presented

with geometric figures varying in their color (red versus blue), shape (circle versus triangle), number of lines (one versus two), position of dot (top versus bottom), and position of gap (left versus right). Two example figures are shown in figure 3.10A. Students then had to guess whether each figure belonged to class A or class B; each time, they were told whether they had guessed correctly or not. Given enough trials, subjects deduced the rule determining class membership: for example, all circular shapes belonged in class A, all triangular shapes belonged in class B, and all the other features were irrelevant.

Once this was mastered, the experimenter changed the rules slightly: Now all figures that were circular *and* had the dot on top belonged to class A, while all figures that were triangular *and* had the dot on the bottom belonged to class B. Subjects could continue to perform well by using their old rule of sorting on the basis of shape; the question was whether they would also learn that the dot position predicted class membership.

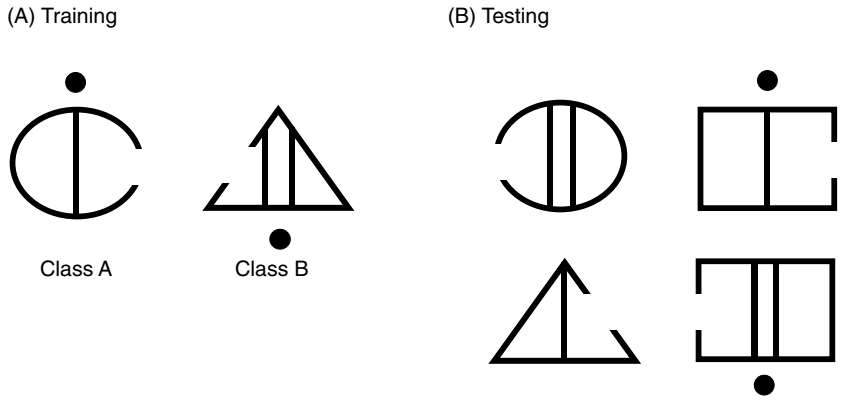


Figure 3.10 Example stimuli used in the human blocking experiment. (A) Subjects were given a set of geometric figures that differed in color (red versus blue in the original experiment), shape (circle versus triangle), number of lines (one versus two), position of dot (top versus bottom), and position of gap (left versus right). In the first phase, all circles belonged in class A and all triangles belonged in class B, regardless of other features. Once subjects acquired this rule, a second feature was made predictive: All figures that were circular *and* had a dot on top belonged in class A, while all figures that were triangular *and* had a dot on the bottom belonged in class B. (Any figures that did not obey this rule were not used.) Since the original rule regarding shape was still predictive, new learning about the dot was blocked. (B) Subjects were given test shapes as shown; given shapes with no dot, subjects were able to accurately sort circles into class A and triangles into class B, demonstrating they had learned the original rule. However, given figures of a new shape (rectangular), subjects were not able to sort on the basis of dot position, demonstrating that learning about the dot had been blocked by prior learning about shape. (Adapted from Trabasso & Bower, 1968, Figure 3.1, p. 74.)

To test this, the experimenter used new figures, shown in figure 3.10B. Given a figure with no dot, all subjects continued to sort the circles into class A and the triangles into class B. However, given a figure with a new shape (rectangular), no subjects were able to correctly sort on the basis of dot position! Thus, prior learning that the shape predicted class membership appeared to have *blocked* subsequent learning that the dot position also predicted class membership. More recent studies have verified that blocking is as pervasive in human learning as it is in animal learning.[18]

The Rescorla-Wagner Model of Conditioning

The blocking effect posed a challenge for simple theories of classical conditioning. It suggested that cues do not merely acquire strength on the basis of their individual relationships with the US; rather, *cues appear to compete with one another for associative strength.* Thus, in the blocking experiment of table 3.1, the light competes with the tone during phase 2; in the case of the Pre-Trained group, the light loses this competition: Since the tone already accurately predicts the US, the light provides no additional predictive information.

This study and others like it led to a growing view among psychologists that to produce effective conditioning, a stimulus cue (CS) must impart reliable information about the expected occurrence of the outcome (US). Moreover, even if a given cue is predictive of a US, it might not become conditioned if its usefulness has been preempted by another co-occurring cue that is a better predictor of the US or that has a longer history of successfully predicting the US. The result of all these findings was to show that "simple" Pavlovian conditioning was not nearly as simple as had once been thought! Instead, animals appear to be very sophisticated statisticians, calculating measures of reliability, usefulness, and predictive value among the stimuli they encounter. In other words, they are acutely sensitive to the **informational value** of stimuli.

It turns out that there is an elegantly simple way to formalize this process developed by Robert Rescorla and Allan Wagner in the early 1970s.[19] The key idea behind the Rescorla-Wagner model is that changes in CS-US associations on a trial are driven by the discrepancy (or error) between the animal's expectation of the US and whether or not the US actually occurred. Sound familiar? Rescorla and Wagner's model of classical conditioning incorporated the same error-correcting principle developed ten years earlier by the engineers Widrow and Hoff. Ironically, it took a decade before two other researchers, trained in both engineering and psychology, realized that the

MathBox 3.2 The Rescorla-Wagner Model

The Rescorla-Wagner model assumes that the change in association between a CS and the US depends on the associative strength of all cues present on that trial. Formally,

$$\Delta w_{ij} = \alpha_i \beta (\lambda_j - V_j) \tag{3.6}$$

where i is a CS and j is a US, w_{ij} is the strength of the existing association between i and j, and Δw_{ij} is the increment to w_{ij} on the current trial. The size of this increment is proportional to three factors: the salience (e.g., amplitude) of CS i (α_i), the learning rate (β), and the difference between the asymptote of conditioning to that US (λ_j) and the total associative strength of all cues present (V_j).

On each trial, the total associative strength is computed as

$$V_j = \sum_i w_{ij} \tag{3.7}$$

for all CS i present on the current trial.

In most cases, the conditioned response CR is simply defined as

$$CR = V_j \tag{3.8}$$

According to the learning rule in Equation 3.6, the changes in associative strength w_{ij} are thus proportional to the difference between the CR and the asymptote λ_j. The asymptote λ_j is generally taken to be 1.0 on those trials in which a particular US j is present and 0.0 on those trials in which no US is present.* Therefore, the learning rule in Equation (3.6) can be restated as

$$\Delta w_{ij} = \alpha_i \beta (US - CR) \tag{3.9}$$

In their original formulation, Rescorla and Wagner noted that the learning rate (β) might differ for different USs; for example, learning might be greater on trials in which the US is present than when it is absent. Additionally, Rescorla and Wagner assumed that CS salience (α_i) might differ for different stimuli. Many later researchers simplified the model by assuming that all CSs in an experiment have equal salience ($\alpha_i = 1.0$), thus,

$$\Delta w_{ij} = \beta (US - CR) \tag{3.10}$$

This is the version of the learning rule that is presented in the text.

Learning in the Rescorla-Wagner model is formally equivalent to the Widrow-Hoff (1960) rule, presented in MathBox 3.1. First, the computation of associative strength (Equation 3.7) can be restated as

$$V_j = \sum_i o_i w_{ij} \tag{3.11}$$

assuming that $o_i = 1.0$ when input i is present and $o_i = 0.0$ otherwise. Thus, this is the same as the activation level computed in the Widrow-Hoff rule (Equation 3.1 in Math Box 3.1).

Second, learning in the Rescorla-Wagner model is mathematically equivalent to the Widrow-Hoff (1960) rule. MathBox 3.1 stated that weight change in the Widrow-Hoff rule was calculated by

$$\begin{aligned} \Delta w_{ij} &= \beta \delta_j o_j \\ &= \beta (d_j - o_j) o_i \end{aligned} \tag{3.12}$$

Again assuming that $o_i = 1.0$ when input i is present and $o_i = 0.0$ otherwise, the rule reduces to

$$\Delta w_{ij} = \beta (d_j - o_j) \tag{3.13}$$

Note that node output o_j is the CR, while the desired output $d_j = \lambda_j = \text{US}$. Thus, the Widrow-Hoff rule can be rewritten as

$$\Delta w_{ij} = \beta (US - CR) \tag{3.14}$$

This is the same as Equation 3.10.

The reader who is interested in further mathematical details, as well as description of the successes and failures of the Rescorla-Wagner model, may refer to Rescorla & Wagner (1972); Wagner & Rescorla (1972); Walkenback & Haddad (1980); Sutton & Barto (1981); Gluck & Bower (1988a); Siegel & Allan (1996).

*In principle, it would theoretically be possible for different USs j to be relatively weaker or stronger, and hence be assigned different values of λ_j.

Widrow-Hoff rule and the Rescorla-Wagner model embodied the same error-correcting principle.[20] MathBox 3.2 provides a formal statement of the Rescorla-Wagner model for the mathematically inclined.

A critical property of the Rescorla-Wagner model of conditioning (and of the Widrow-Hoff training rule for neural networks) is that the weights associated with one cue can indirectly influence the weights accruing to other co-occurring cues. That is, if a tone and light are both present on a trial, they will compete for associative weight. This competitive property of the Rescorla-Wagner model allows it to account for many important conditioning phenomena, especially those with complex stimuli involving the presentation of multiple stimulus elements (such as tones and lights paired together).

For example, the Rescorla-Wagner model (and the Widrow-Hoff rule) can account for the blocking effect. Consider the network in figure 3.11A; this represents the state of affairs after phase 1 training in which a tone predicts an airpuff US. Now, at the start of phase 2, the light and tone are presented

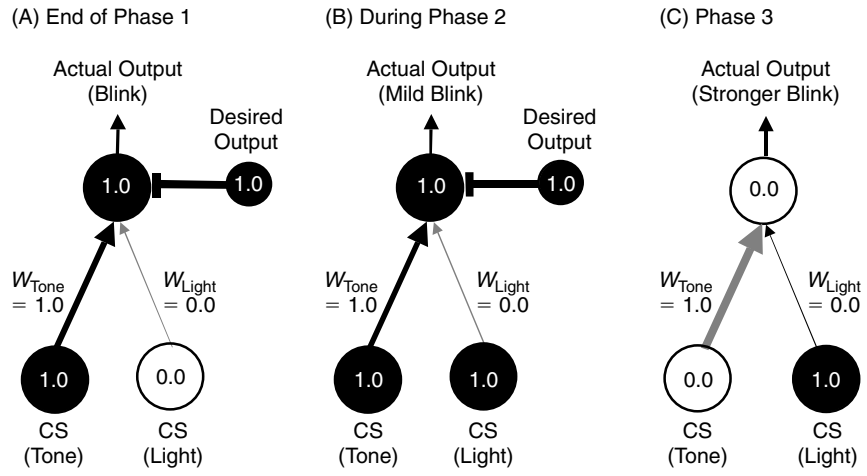


Figure 3.11 Blocking in the Rescorla-Wagner model. (A) Phase 1 involves repeated pairing of the tone CS with an airpuff US. At the end of this phase, there is a strong weight from the tone input node, which in turn can activate the output node. (B) Phase 2 involves presentation of both tone and light CSs, paired with the US. Since the tone already perfectly predicts the US and the output node is fully activated, there is no output error, and no learning occurs. Hence, the weight from the light input node remains at 0.0. (C) Finally, in phase 3, the network is tested with presentations of light alone. Since the weight was never changed from 0.0, there is no activation of the output node, and no behavioral response occurs. Hence, the Rescorla-Wagner model correctly shows blocking.

together (figure 3.11B). The output node receives activation from both input nodes, multiplied by their weights: $1.0 \times 1.0 = 1.0$ from the tone and 1.0×0.0 from the light for a total of 1.0. This is the output node's activation. Since the airpuff US is present on this trial, the response is correct—and the error is 0.0. Since weight change is determined by the error and the error is 0.0, there is no learning on this trial—or, indeed, on any other phase 2 trial. By the end of phase 2, the network weights will still be the same as they were at the beginning of phase 2: The tone is strongly weighted, and the light is not. A subsequent test presentation of the light in phase 3 is shown in figure 3.11C: There is no weight from the light input node, so there is no activation of the output node—so there is no response to the light alone. Thus, the network will show blocking—just like the Pre-Trained animals in figure 3.9.

The blocking effect is only one of numerous subtle conditioning phenomena that are addressed by the Rescorla-Wagner model. Later in this chapter, we will return to the blocking effect and the way in which it has helped to elucidate the neural mechanisms of conditioning.

Broad Implications of the Rescorla-Wagner Model

One of the most important lessons to be learned from the Rescorla-Wagner model is that it is possible to understand a large body of data via a relatively simple mathematical rule. An equally important implication is that learning in many different experimental procedures and many different species—including humans—shares some fundamental characteristics.

A quarter century after its publication, the Rescorla-Wagner model is generally acknowledged as the single most powerful and influential formal model of learning ever produced by psychology.[21] The model gained such wide acceptance because it was simple, it was elegant, and could explain a wide range of previously puzzling empirical results. This is one hallmark of a successful model: *The model should show underlying order among a series of results initially seem unrelated or even contradictory.*

The Rescorla-Wagner model also made novel and surprising predictions about how animals ought to behave in new experimental procedures, and experimenters rushed to test these predictions. This is another feature of a successful model: *It should generate subtle predictions that were not intuitively obvious before.* Ideally, modeling and empirical work should cycle; the model should make predictions that, when tested, provide new data. If the data match the predictions, the model is supported. If not, then the model must be revised. This revised model then generates new behavioral predictions, and the cycle begins again.

By virtue of its simplicity, the Rescorla-Wagner model does not account for all kinds of learning. Many researchers devoted themselves to showing how one or another addition to the model allows it to account for a wider range of phenomena[22]—but with too many additions, the model loses some of its simplicity and appeal. In the next chapter, we will focus on some important limitations of the model and mechanisms that have been proposed to circumvent these limitations. However, these limitations need not be considered shortcomings of the Rescorla-Wagner model *per se*. Rather, these limitations are more appropriately understood as lying outside the model's stated domain.

Within its domain, the Rescorla-Wagner model combines explanatory power with mathematical simplicity. It takes an intuitively tractable idea—namely, that learning is driven by error—pares away all but the most essential details, and explores implications of this idea that were not obvious before. The Rescorla-Wagner model is also a starting point from which many subsequent models were built; some of these newer models will be described in the chapters to come.

Error-Correction Learning and the Brain

Currently, most neuroscientists believe that an important mechanism of biological learning is the alteration of synaptic strength. There is an obvious parallel between this idea and the idea of weight adjustment in the Rescorla-Wagner model. So it is tempting to jump to the conclusion that the Rescorla-Wagner model describes a rule for synaptic changes between individual neurons.

However, there is good evidence that this is *not* the case. The Rescorla-Wagner model requires a teaching signal, to provide information about the desired output. If there were more than one output node, as could happen in a more complicated network, there would need to be a different teaching signal for each output node. The brain, with its immense number of neurons, would require an equally staggering number of teaching signals if it were to implement the Rescorla-Wagner model of learning at each synapse.

Our current understanding of the way in which neurons form and alter connections is incomplete. An extensively studied candidate mechanism is **long-term potentiation** (**LTP).**[23] In one form of LTP (called **associative LTP**), simultaneous activity in two neurons leads to a strengthening of the synapse that connects them to each other. This learning process was anticipated by Donald Hebb in the 1940s, long before the technology existed to observe activity in individual neurons,[24] and the basic idea harks as far back as the

psychologist William James in the nineteenth century.[25] LTP is a mechanism for synaptic change that depends on neuronal correlation but does not require a teaching input. Thus, this form of synaptic learning (also known as *synaptic plasticity*) does not implement error-correction as embodied in the Rescorla-Wagner model.

Nevertheless, while the Rescorla-Wagner model of learning probably does not take place at the cellular level, error-correction mechanisms do appear to emerge from brain circuits. Certain brain regions have very strong inputs that do appear to act as teachers, specifying desired outputs for the entire region, if not for individual cells. One example is the cerebellum, an essential brain region for the storage and expression of learned motor reflexes, such as classical eyeblink conditioning.[26]

Figure 3.12 shows a very simplified schematic of the cerebellar pathways for eyeblink conditioning, based on the work of Richard Thompson and colleagues. Information about a CS, such as a tone or light, enters through primary sensory processing areas in the brain and eventually reaches the cerebellum. Information about a US, such as a corneal airpuff, travels through a way station known as the **inferior olive.**

Climbing fibers carry this information from inferior olive to cerebellum, where they make very powerful synapses onto cerebellar neurons. The output from the cerebellum is a signal that is related to the strength of the conditioned eyeblink response; this information travels to brain structures that are responsible for executing the motor commands to close the eyelid.* Conditioned response information also travels along a feedback pathway to inhibit the inferior olive.

The feedback pathway from cerebellum to inferior olive is inhibitory, which means that when a conditioned eyeblink response is generated, inferior olive activity is depressed. The other input to the inferior olive, carrying US information, is excitatory. Various neurophysiological studies have suggested that the inferior olive activity reflects the difference between US and the conditioned response, which can be formalized as

$$\text{InferiorOliveActivity} = \text{US} - \text{CR}$$

Note that the US here is equivalent to information about the desired output. When the US is present, the animal should have blinked. When the US is not present, the animal should not have blinked. Thus, this subtraction of US and

*In fact, the relationship between cerebellar output and CR generation is complicated, since the output of cerebellar Purkinje neurons is inhibitory; CRs are generated when Purkinje cells decrease their activity.

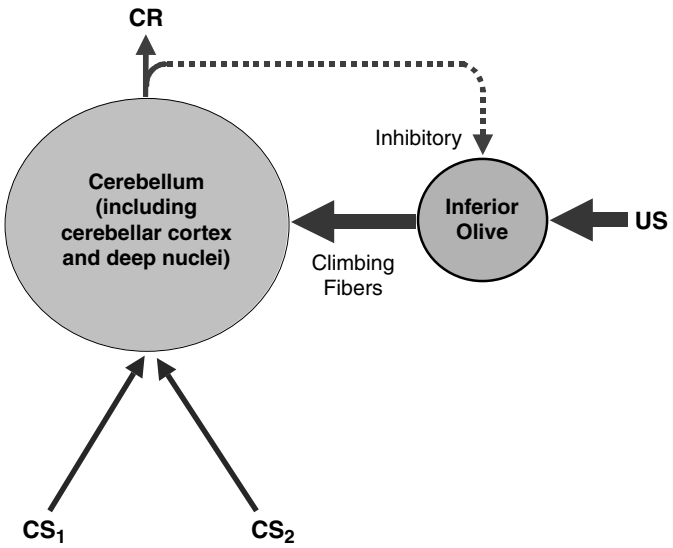


Figure 3.12 The cerebellum (including cerebellar cortex and deep cerebellar nuclei) receives information about auditory CSs via a connection from early sensory processing centers in the brain. It also receives information about airpuff USs via a pathway that passes through the inferior olive. The inferior olive also receives an inhibitory feedback connection from the cerebellum. As the CR develops, inhibition increases in the inferior olive, reducing the amount of US information that reaches cerebellum. In effect, the output of the inferior olive is proportional to (US – CR). This is analogous to the error term in the Rescorla-Wagner model. Thus, the inferior olive could provide the error signal to drive cerebellar learning in motor-reflex conditioning. The cerebellar-inferior olive system may implement error correction learning in this circuit.

conditioned response information is essentially the same as the error computation in the Rescorla-Wagner model, namely,

$$\begin{aligned}\text{OutputError} &= \text{DesiredOutput} - \text{ActualOutput} \\ &= \text{US} - \text{CR}\end{aligned}$$

In fact, inferior olive activity during acquisition starts off high, before the conditioned response is reliably generated, and gradually drops back to baseline as the conditioned response is acquired (figure 3.13A,B).[27] This activity pattern is basically the same as the pattern of output error generated by the Rescorla-Wagner model during learning (figure 3.13C,D). This suggests that the inferior olive could be computing output error, providing the information necessary to implement error-correction in the cerebellum.

Once the error is computed, the next step in the Rescorla-Wagner model is to update weights accordingly, to reduce the error on subsequent trials. To do this, the error signal must be transmitted to the connections between input

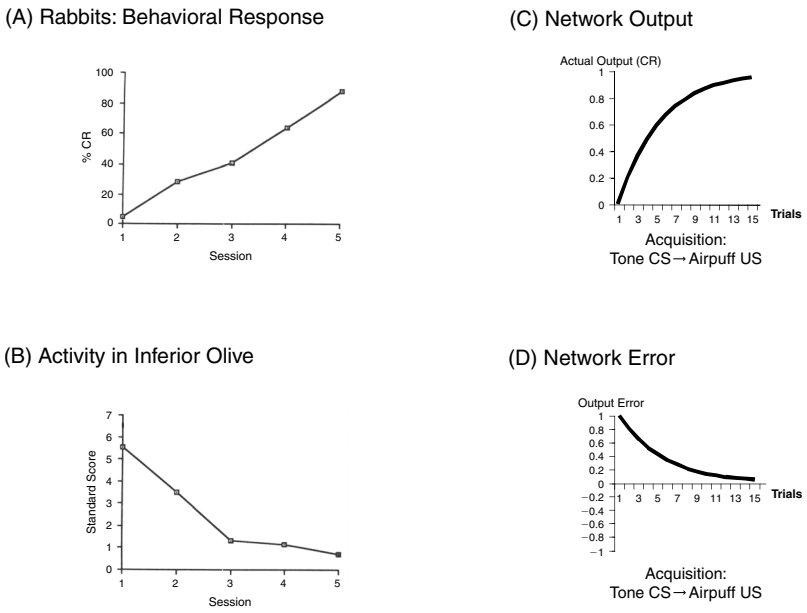


Figure 3.13 While a rabbit is acquiring a conditioned eyeblink response (A), measurements are taken from the inferior olive (B). Activity in the inferior olive is high early in training, when the CR is low, and decreases as the CR develops. Activity in the inferior olive is given as standard scores, representing the activation during the presentation of the CS relative to baseline activity. The decrease in inferior olive activity as a function of training is similar to the decrease in network error in the Rescorla-Wagner model (D) as a function of learning (C). This has led to the suggestion that the inferior olive might be calculating response error. (A and B are adapted from Sears & Steinmetz, 1991, Figure 3.)

and output nodes. In the cerebellum, the same neurons that receive sensory information are also contacted by climbing fibers, carrying information from the inferior olive (figure 3.12). Synapses carrying sensory information to the cerebellar neurons are known to be modified depending on the presence or absence of climbing fiber activity.[28]

On the basis of these data, we have proposed a model of the cerebellum as a circuit for implementing the Rescorla-Wagner model of learning.[29] The basic idea is as follows: When a tone CS is presented, sensory information reaches the cerebellum, but no conditioned response is elicited. If the US appears, this information travels to the inferior olive, which computes the error as US − CR = 1.0 − 0.0 = 1.0. This error information is carried via climbing fibers to the cerebellar neurons, and initiates synaptic change. Over many

CS-US pairings, the net effect is to make the cerebellum more likely to generate a response when the CS appears.[30] As the conditioned response begins to emerge, the feedback pathway from cerebellum to inferior olive becomes active. This inhibits activity in the inferior olive. Eventually, when the response is completely learned, there is no activity in the inferior olive (since US − CR = 1.0 − 1.0 = 0.0). At this point, the climbing fibers are silenced, and learning stops. Thus, error-correction might be implemented at the level of the cerebellar-inferior olive circuit.

The power of the simple cerebellar model shown in figure 3.12 can be illustrated by its application to behaviors such as blocking.[31] As was described earlier, a blocking procedure involves several phases. In phase 1, a cue such as a tone predicts an airpuff US (see table 3.1). This is followed by phase 2, in which a tone-light compound is paired with the US. In phase 3, there should be little or no response to a test of the light. This, according to the Rescorla-Wagner model, is because prior learning to the tone blocked learning about the second stimulus. The cerebellar model of figure 3.10 expects that, during phase 1, the tone will eventually come to elicit a conditioned response, at which point the output error is 0, and inferior olive activity will be fully inhibited. When tone-light conditioning begins, there is still no error—since the tone elicits a strong conditioned response—and so there is no learning about the light.

This interpretation assumes that blocking depends on low output error, and hence low inferior olive activity, during phase 2. Thus, blocking should be disrupted if inferior olive activity is inflated during phase 2. A recent study by Thompson and colleagues confirmed this prediction.[32] In their study, rabbits were first trained to give reliable eyeblink responses to a tone CS. Next, the rabbits were given injections of a drug (picrotoxin) that temporarily disabled the feedback pathway from cerebellum to inferior olive (figure 3.14). With this pathway disabled, the inferior olive's activity should reflect the US, without any information about the conditioned response:

InferiorOliveActivity = US − 0

The rabbits were then given phase 2 training of blocking, in which the compound of tone CS and light CS was paired with the US. In these rabbits, inferior olive activity was high whenever the US was presented, whether or not a conditioned response was generated. Thus, learning continued; in phase 3, the rabbits gave a strong response to the light CS: Blocking had been prevented.

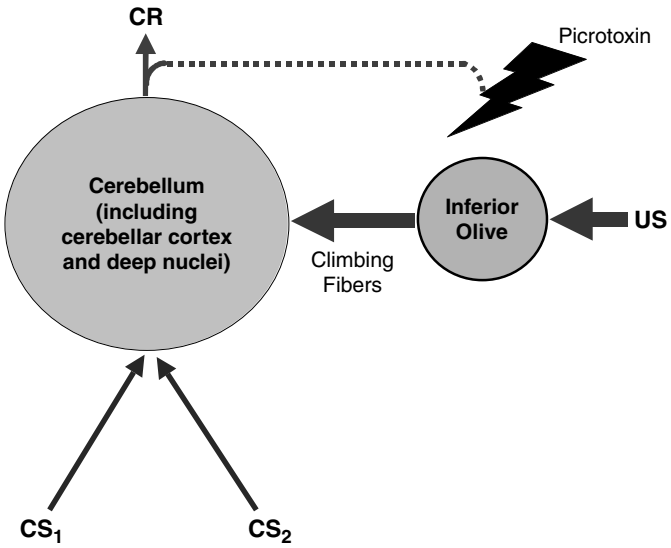


Figure 3.14 The cerebellar circuit model of figure 3.12 predicts that error-correction learning depends on the inferior olive computing the difference between US and CR. If the CR feedback pathway is interrupted, then the inferior olive output should always mirror the US, regardless of the CR. Blocking and other behaviors that depend on an error signal should also be disrupted. In fact, when the CR feedback pathway is disrupted by injection of the drug picrotoxin to the inferior olive, blocking is indeed disrupted in rabbit eyeblink conditioning.

These and related results provide strong support for the interpretation of the cerebellar-inferior olive circuit as implementing error correction.[33] The cerebellum is not the only place in the brain where such error-correction learning may take place; just as the strong climbing fibers may serve as teaching inputs to cerebellum, there may be analogous systems in the hippocampus and elsewhere. For example, similar error-correcting circuits may exist in the brain structures that are responsible for learning fear responses to threatening stimuli.[34] Brain chemicals, such as dopamine, may also play a role in broadcasting error signals throughout the brain.[35]

Still, error correction, as assumed in the Rescorla-Wagner model, may be the exception rather than the rule in brain systems. Teaching inputs may have evolved to serve very specialized functions in a few critical brain regions. It is important to note that error correction is only one behavioral principle of learning; other brain mechanisms may be devoted to implementing other principles, and some of these will be introduced in later chapters. The next, and final, section of this chapter, reviews some of the limitations of error-correction learning.

3.4 LIMITATIONS OF ERROR-CORRECTION LEARNING

In the Rescorla-Wagner model, all weight changes depend on the prediction error at the output node. This means that learning in the network is governed by the degree to which network output (conditioned response, or CR) deviates from desired output (US). No other learning occurs. Thus, if the network is simply exposed to a stimulus, without any US, then US = CR = 0, output error is zero, and there is no learning. However, several studies with animals and humans suggest that individuals can learn about stimuli just from exposure to those stimuli, without any explicit reinforcement (US) or response (CR). This general class of phenomena—often called perceptual learning, or learning through mere exposure—can be accommodated in neural networks only by making additional assumptions about the network architecture or the learning algorithm. In chapters 5 and 6, we will review some of these possible solutions and argue that they are relevant to understanding different kinds of learning in the brain. For now, we will discuss two simple conditioning procedures that involve learning without explicit reinforcement.

Sensory Preconditioning

Back in chapter 2, we introduced a behavioral phenomenon called **sensory preconditioning.** Table 3.2 reviews the experimental procedure, using the example of eyeblink conditioning. In sensory preconditioning, rabbits are assigned to one of two groups. In the Compound Exposure group, rabbits first receive unreinforced exposure (that is, no US) to tone and light stimuli, presented together in a compound. In the Separate Exposure group, rabbits receive equivalent unreinforced exposure to the tone and light, presented separately. In phase 2, all rabbits are then trained that the light predicts a blink-evoking airpuff US. Once a conditioned response is learned to the light, all rabbits are then tested for their response to the other, tone stimulus. Rabbits in the Separate Exposure group give little or no response to the tone. By contrast, rabbits in the Compound Exposure group give a significant response to

Table 3.2 Sensory Preconditioning

Group	Phase 1	Phase 2	Test
Compound Exposure	Tone and light (together); no airpuff	Light → airpuff ⇒ **blink!**	Tone ⇒ **blink!**
Separate Exposure	Tone, light (separately); no airpuff	Light → airpuff ⇒ **blink!**	Tone ⇒ no blink

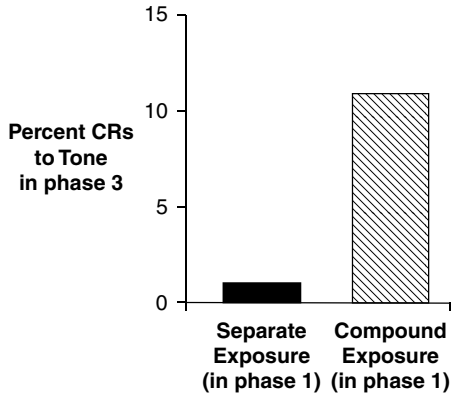


Figure 3.15 Sensory preconditioning in rabbit eyeblink conditioning. One group of rabbits is given exposure to the tone and light, separately, in phase 1. These rabbits are then trained to respond to the light in phase 2. In phase 3, the rabbits are presented with the tone and show very little transfer of learned responding. A second group of rabbits is given phase 1 exposure to the tone and light, presented together as a compound stimulus. These rabbits are then trained to respond to the light; when tested with the tone in phase 3, they give a strong response. (Plotted from data presented in Port & Patterson, 1984.)

the tone (figure 3.15). In effect, prior exposure to the tone-light compound causes these two stimuli to be "chunked" or compressed; later learning to one stimulus then tends to generalize strongly to the other stimulus.

Now consider what happens with a network, such as the one in figure 3.6, given the same training. Initially, all network weights are set to 0.0. This means that the network will not respond to any stimulus or combination of stimuli. During phase 1 (the exposure phase), if the network is given exposure to the tone-light compound, it will not generate a response. Since there is no US during phase 1, this nonresponse is correct: There is no output error, and so there is no learning. The weights do not change during this phase. Thus, pre-exposure in the absence of reinforcement has no effect on the network. The weights at the beginning of phase 2 are the same for a network given compound exposure as for one given separate exposure—or no exposure at all. In all three cases, phase 2 learning about the light transfers minimally to the tone. Thus, the Rescorla-Wagner model does not account for sensory preconditioning.

Latent Inhibition

Another example of learning through mere exposure is **latent inhibition,** which has been developed and extensively studied by Robert Lubow.[36] Table 3.3 schematizes a latent inhibition procedure, again using a rabbit

Table 3.3 Latent Inhibition

Group	Phase 1	Phase 2
CS-Exposure	Tone CS	Tone CS → airpuff US **. . . Slow learning**
Sit-Exposure	*(Sit in experimental chamber)*	Tone CS → airpuff US **. . . Normal learning**

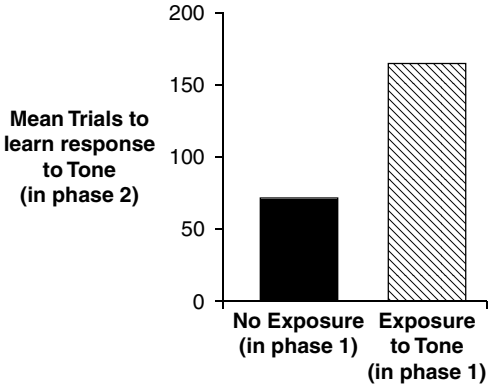


Figure 3.16 Latent inhibition in rabbit eyeblink conditioning. One group of rabbits is given exposure to the tone, while a second group is given an equivalent amount of time in the experimental chamber, but not exposed to the tone. Next, both groups of rabbits are trained to respond to the tone. The nonexposed rabbits learn the association quickly; the exposed rabbits are significantly slower to learn. (Plotted from data reported in Solomon & Moore, 1975.)

eyeblink conditioning example. Rabbits in the CS-Exposure group are given phase 1 presentations of a CS (e.g., a tone) without any reinforcement. For comparison, rabbits in a Sit-Exposure group are allowed to sit in the conditioning chamber for an equivalent amount of time with no exposure to any CS. In phase 2, both groups receive training in which the tone is paired with the US until the rabbit gives a reliable conditioned response. Rabbits in the Sit-Exposure group learn this response at normal speed; rabbits in the CS-Exposure group, previously exposed to the tone, show much slower learning, as is seen in figure 3.16.

Latent inhibition is a complex and intriguing phenomenon that may reflect the contribution of many different processes in the normal brain. We will discuss some of these in chapters 7 and 9. For now, we just note that the Rescorla-Wagner model learns only when there is output error, that is, when the CR does not equal the US. In phase 1 of latent inhibition, the CR is

initially 0—and there is never any US, so the output error is always 0. Thus, both the CS-Exposed and Sit-Exposed conditions undergo no learning in phase 1 and hence are expected by the Rescorla-Wagner model to perform identically in phase 2. Thus, the Rescorla-Wagner model fails to capture the robust and widely demonstrated behavior of latent inhibition.

Implications of the Limitations of Error-Correction Learning

Section 3.3 argued that the error-correction principle of the Rescorla-Wagner model is a powerful model that can account for a wide range of conditioning behaviors in a relatively simple and elegant way. Further, it appears to have some biological relevance: Cerebellar circuitry is sufficient to implement the kind of error computations required by the model and to use error signals to drive learning. Thus, the Rescorla-Wagner model may be more than a psychological description of learning; it may be a model of how the cerebellum implements conditioned learning.

However, the Rescorla-Wagner model has several limitations, most notably its failure to account for mere exposure effects such as latent inhibition and sensory preconditioning. It is interesting to note that sensory preconditioning and latent inhibition are among the behavioral paradigms that are disrupted by damage to the hippocampal-region.[37] This suggests that something about mere exposure effects might be especially sensitive to hippocampal-region mediation, while other effects—such as those captured by the error-correcting principle of the Rescorla-Wagner model—might not depend on hippocampal-region mediation but, rather, depend on the cerebellum and other extrahippocampal structures.

Thus, it might be possible to model a full range of conditioned behaviors by considering a system that has a "cerebellar" module, which implements simple error-correction learning, and a "hippocampal" module, which includes other capabilities such as mere exposure learning and representational compression.

This leads to the question of what kind of a model the hippocampal module might be. Clearly, it must have additional capabilities beyond error correction. Perhaps the chief requirement is the ability to form new stimulus representations: ways of representing stimulus information independent of their association with the US. The next chapter, chapter 4, explores how these representational issues can be addressed through network models that are elaborations of the error-correction networks presented in this chapter.

SUMMARY

• Most neuroscientists now believe that the basic mechanism of learning is alteration of synaptic strength.

• The power of nodes in a neural network (and neurons in the brain) comes not just from their individual power, but from their collective power, which emerges only when many such devices are interconnected in a network.

• The key to learning in neural networks is changing the weights.

• Researchers in both psychology (Rescorla and Wagner) and engineering (Widrow and Hoff) argued that weights should be changed proportionally to output error, the difference between actual and desired responses. A teaching signal provides information about the desired response. This kind of learning is called error-correction learning, because over many trials, it tends to minimize the output error.

• The Rescorla-Wagner model (and the Widrow-Hoff rule) can account for some simple forms of learning, including acquisition of a conditioned eyeblink response, and more complex training procedures, such as blocking. It does not account for results such as latent inhibition and sensory preconditioning, which involve learning after mere exposure to stimuli in the absence of reinforcement (such as a US).

• There is evidence that error correction takes place in some places in the brain. For example, during eyeblink conditioning, the inferior olive may provide a teaching signal to allow cerebellar neurons to reduce the difference (error) between their output (the CR) and the desired response (a prediction of the US). This is a circuit-level instantiation of the Rescorla-Wagner model. Error-correction circuits may also underlie some fear learning. However, error correction may be the exception rather than the rule in brain systems.