COMPARATIVE
COGNITION & BEHAVIOR
REVIEW
A peer-reviewed monograph published by Comparative Cognition & Behavior Reviews on behalf of the Comparative Cognition Society.

Connectionism and Classical Conditioning

Michael R.W. Dawson

University of Alberta
Edmonton, Alberta
Canada T6G 2P9
1-(780)-492-5175

Abstract

The purpose of this monograph is to examine the relationship between a particular artificial neural network, the perceptron, and the Rescorla-Wagner model of learning. It is shown that in spite of the fact that there is a formal equivalence between the two, they can make different predictions about the outcomes of a number of classical conditioning experiments. It is argued that this is due to algorithmic differences between the two, differences which are separate from their computational equivalence.

mdawson@ualberta.ca

Rosenblatt Software Available As Freeware At:
http://www.bcp.psych.ualberta.ca/~mike/Software/Rosenblatt/index.html

A peer-reviewed monograph published by Comparative Cognition and Behavior Reviews on behalf of the Comparative Cognition Society. doi: 10.3819/ccbr.2008.30008

Chapter 1: Learning in Networks and Animals

The purpose of this book is to explore the relationship between a particular artificial neural network, the perceptron, and a particular theory of animal learning, the Rescorla-Wagner model. This chapter introduces the need for this exploration. First, it provides an example empirical observation that demonstrates that perceptrons do not always behave in the way that the Rescorla-Wagner model of animal learning predicts, even though the two are presumed to be formally equivalent. Second, in examining formal treatments of the relationship between network and animal learning, it is revealed that extant proofs do not provide what is usually thought to be the case. These two observations motivate a detailed empirical and formal comparison of perceptron learning to animal learning. The main goal of this chapter is to provide the plan for how this comparison proceeds, and to highlight general themes that are important to it.

A peer-reviewed monograph published by Comparative Cognition and Behavior Reviews on behalf of the Comparative Cognition Society.

1.1 An Empirical Puzzle

1.1.1 Identical, But Different

There is considerable interest in the relationship between models of animal learning and models of learning in artificial neural networks. For instance, one can easily find many simulations of associative learning experiments conducted with neural networks (e.g., Delamater, Sosa & Katz, 1999; Carpenter, 2001; Enquist & Ghirlanda, 2005). French, 1999; Grossberg,1980; Kehoe, 1988; Kruschke, 1992, 1996a, 1996b, 2001; Mirman, & Spivey, 2001; Nickerson et al, 2006; Pearce, 2002; Rizzuto & Kahana, 2001; Roitblat & von Fersen, 1992; Schmajuk & Blair, 1993; Schmajuk & DiCarlo, 1992; Schmajuk, Lam & Gray, 1996; Schmajuk, Lamoureux & Holland, 1998; Shanks, 1995; Sutton & Barto, 1998; Yaremchuk, Willson, Spetch, & Dawson, 2005).

How are such simulations justified? At a very general level, artificial neural networks implement very general theories of association (e.g., Bechtel, 1985; Shanks, 1995), and thus appear to be able to inform theories of associative learning that have been developed in other domains, such as comparative psychology. At a more technical level, it can be shown that the rule used to train a particular type of network, called a perceptron, is formally equivalent to a foundational theory of animal learning, the Rescorla-Wagner model (Sutton & Barto, 1981). "That the Rescorla-Wagner equation was developed to account for animal learning behavior, whereas the nearly identical Widrow-Hoff rule was formulated to approximate the solutions of sets of linear equations, suggests that these rules describe some ingredient essential for adaptive behavior" (Sutton & Barto, 1981, p. 165). In short, formal analyses indicate that some theories of animal learning are identical to particular theories of neural network learning.

The formal equivalence between perceptron learning and the Rescorla-Wagner model has become a textbook staple (e.g., Gluck & Myers, 2001; Pearce, 1997). However, if one takes the trouble to use a perceptron to simulate standard paradigms in classical conditioning, then one can discover a number of instances in which the artificial neural network generates different predictions than does the Rescorla-Wagner model (e.g., Dawson & Spetch, 2005). That is, while the two are identical at a formal level, they can be different still be different at another level of analysis.

1.1.2 Networks and Learning

The surprising dissociation between network learning and animal learning provided the motivation for this book. The purpose of this book is to provide a careful examination of how learning in perceptrons can be related to simple kinds of associative learning that have long been studied in animals.

Some of this exploration is empirical in nature. That is, a number of different classical conditioning paradigms are simulated using perceptrons, sometimes manipulating some of the basic characteristics of the neural networks (in particular, by changing the activation function that the output units of these networks use to convert an incoming signal into an observable response). These simulations are intended to explore what kinds of classical conditioning phenomena that have been observed in the animal literature can be modeled by these simple networks, as well as to determine what phenomena may not be captured by them. Furthermore, these simulations are aimed at uncovering discrepancies between the performance of the networks and the predictions of the Rescorla-Wagner model.

Some of this exploration is formal in nature. If the two different traditions of modeling make different predictions, then is it possible that claims about their formal equivalence are mistaken? If this is not the case, then how is it possible those two systems can be identical from one perspective, but at the same time be different from an alternative point of view?

1.2 A Formal Surprise

1.2.1 Perceptron Nonlinearity

A perceptron is a simple artificial neural network that responds to presented stimuli (Rosenblatt, 1962). For example, two input units can represent the presence or absence of two conditioned stimuli. A single output unit can represent the system's response to these stimuli. The network is trained to respond correctly by providing it feedback about its responses to different stimuli. Training modifies the weights that connect input units to the output unit, which are analogous to associative strengths.

Importantly, perceptrons have an intrinsic nonlinearity built into their output units. This nonlinearity defines how input signals are converted into output responses.

When a perceptron is presented a stimulus, its first task is to compute a signal called the net input. The net input is a linear combination of stimulus values (i.e., whether some conditioned stimulus is present or not) with the current associative strengths of the stimuli. We will see later that net input is identical to the overall association term (ΣV) in the Rescorla-Wagner model.

After net input is computed, a perceptron's second task is to convert it into a response value. This is typically accomplished using a nonlinear equation called an activation function. Two of the earliest networks, the perceptron (Rosenblatt, 1962) and ADALINE (Widrow, 1962; Widrow & Hoff, 1960) used the Heaviside equation as an activation function. That is, the output unit generated one number (e.g., 0) to indicate that it was off, and a second number (e.g., 1) to indicate that it was on. No other response values are produced. A more modern variant of these machines, which we will call an integration device, approximates the Heaviside equation with a sigmoid-shaped function produced by the logistic equation (Rumelhart, Hinton, & Williams, 1986). Another uses a bell-shaped activation function defined by the Gaussian equation (Dawson & Schopflocher, 1992).

Each of networks from the preceding paragraph is an example "perceptron". One of the characteristics of this class is that it has no intermediate processing units ("hidden units") that stand between the input and output units. A second characteristic is that the output unit's activation function is nonlinear.

1.2.2 A Linear Proof

The generic rule for training a perceptron, the delta rule, is an example of a least mean squares (LMS) method of reducing error. A typical statement of how such a rule relates to animal learning is the "LMS rule is essentially identical to the Rescorla-Wagner model" (Gluck & Bower, 1988, p. 230). Similarly, Quinlan (1991, p. 56) notes "that a special case of the delta rule is formally equivalent to the Rescorla-Wagner rule." Sutton and Barto (1981, p. 155) point out that the Widrow-Hoff rule is "essentially identical" to the Rescorla-Wagner model, but later make the stronger claim that "these two models are, in fact, identical (p. 156).

These quotes qualify the relationship between the two models ("essentially identical", "special case") because proofs of the equivalence between the two assume that the networks are linear (Gluck & Bower, 1988; Gluck & Myers, 2001; Sutton & Barto, 1981). That is, the response of an output unit is assumed to be its net input, rather than a nonlinear transformation of the net input. In other words, the proofs do not show a formal relationship between perceptrons proper and the Rescorla-Wagner model.

1.2.3 A Nonlinear Proof

That perceptrons have not been proven to be identical to the Rescorla-Wagner model might explain the surprising result from Section 1.1. However, later in this book this possibility is removed. This is because a proof of the formal equivalence between nonlinear perceptrons and the Rescorla-Wagner model is provided. A key element of the proof – relating net input to response – is important, though, in explaining the dissociation between network and animal learning.

1.3 Cognitive Science and Associative Learning

1.3.1 Cognition vs. Association

It has been argued that one of the characteristics of cognitive science is its adoption of the tri-level hypothesis (Dawson, 1998). According to this hypothesis, three different questions have to be answered to explain an agent's information processing (Marr, 1982; Pylyshyn, 1984). At the computational level, one answers the question "what information processing problem is being solved?" At the algorithmic level, one answers the question "what processing steps are employed to solve the problem?" At the implementational level, one answers the question "what mechanisms are responsible for carrying out the information processing steps?"

One of the interesting features of the cognitive revolution that began in psychology in the 1950s is its strong bias against associationism. This began with cognitive psychology's birth as a reaction to behaviorism (Chomsky, 1959), and continued with more general claims that associationist models were not powerful enough to provide accounts of such phenomena as language (Bever, Fodor, & Garrett, 1968).

One of the consequences of the artificial neural network resurgence that occurred in the 1980s (McClelland & Rumelhart, 1986; Rumelhart & McClelland, 1986) was a resurgence of interest in relating human cognition to association (Bechtel, 1985). Not surprisingly, this led to many of the old cognitive criticisms of associationism being restated as criticisms of the new connectionism (Fodor & Pylyshyn, 1988; Lachter & Bever, 1988; Pinker & Prince, 1988).

However, many would argue that these new models have an important place within cognitive science. The tri-level hypothesis can easily be applied to artificial neural network models, substantially blurring the differences between them and symbol-based cognitive theories (Dawson, 1998; Dawson, Medler, & Berkeley, 1997; Dawson, Medler, McCaughan, Willson, & Carbonaro, 2000). Similarly, it has been argued that the tri-level hypothesis can be usefully applied to associative learning in general (Shanks, 1995).

1.3.2 Levels and Equivalence

When applied, the tri-level hypothesis reveals issues that are crucial to the relationship between network and animal learning. In particular, there is a one-to-many relationship from one level to another (Fodor, 1975). That is, any one information processing problem described at the computational level can be solved by more than one different procedure described at the algorithmic level. In short, different algorithms can solve the same problem.

One consequence of this is that two systems might be equivalent at one level, but differ at another, when the tri-level hypothesis is applied to them. For instance, two systems might be equivalent in the sense that they are saving the same problem. However, measures of their responses might lead to different results if the two are using different algorithms to solve this problem.

One of the themes of this book is that the apparent paradox that network learning is formally equivalent to animal learning but produces different empirical predictions can be unraveled by paying attention to different levels of analysis. This point is often neglected when neural networks are used to study associative learning. We will see that researchers will argue against using a particular type of network because it is formally equivalent to the Rescorla-Wagner model, and therefore must inherit its limitations (Pearce, 1997). In their assessment of the relationship between network and animal learning, Sutton and Barto (1981, p. 156) noted that "the limitations of linear learning rules are not as devastating as once thought." We will see that this is so is because network algorithms might actually repair some failures of animal models that are formulated at the computational level.

1.4 The Synthetic Approach

1.4.1 The Analytic Approach

Most models in cognitive science and in experimental psychology are derived by analyzing existing behavioral measurements (Dawson, 2004). The result, which is usually either a model of data or a mathematical model, is designed to summarize these measurements, and is almost always evaluated in terms of its goodness of fit to extant data.

This approach to modeling has been called the *analytic approach* (e.g., Braitenberg, 1984; Pfeifer & Scheier, 1999). This is because when it is employed models emerge from the analysis of existing data. In the analytic approach, data precede models.

1.4.2 A Synthetic Alternative

The analytic approach has been enormously successful, and is deserving of its methodological dominance. However, it is not the only approach available to researchers.

An alternative methodology has been called the synthetic approach. In general, practitioners of the synthetic approach make assumptions about basic properties of systems or agents. These primitive capacities are then built into working systems – models that might be computer simulations or robots – whose behavior is then observed. What kinds of behaviors emerge from the working system, and what kinds of behaviors don't emerge? Furthermore, do any behaviors arise that are more complicated than what one might predict from knowledge of the components that were used to create the system? In short, in the synthetic approach, model construction *precedes* behavioral analysis.

The synthetic approach has emerged from the study of behavior-based robotics (e.g. Brazeal, 2002; Brooks, 1999; Webb & Consi, 2001), from whence has become enormously influential. Most developments that have occurred in the general field of embodied cognitive science (e.g., Agre, 1997; Clark, 1997; Varela, Thompson & Rosch, 1991; Wilson, 2004) can be traced back to the synthetic approach in robotics, which in turn has its roots in pioneering work in cybernetics (e.g., Ashby, 1960; Grey Walter, 1963; Wiener, 1948). Braitenberg (1984) has argued that the synthetic approach is likely to produce simpler theories or models than those developed within the more popular analytic tradition.

1.4.3 Synthetic Association

Another perspective on the current manuscript is that it provides an example of the synthetic approach. Dawson (e.g., 2004) has argued that artificial neural networks provide an ideal medium for exploring psychological phenomena synthetically. Many of the chapters that follow illustrate this point by showing how very simple artificial neural networks, which incorporate a very small number of simple component processes, can be used to explore associative learning.

One consequence of this is that the simulations studies that follow are intended to explore the kinds of responses that can be generated by different types of simple networks, as well as to explore the changes in responses that result when different assumptions are used to build the networks (e.g., by changing the activation function used by output units to generate responses). There is no doubt that the relationship between model responses and behavioral data is important, and this relationship is evident throughout the manuscript. However, from the synthetic perspective detailing this relationship is less important than exploring the responses that emerge from the family of networks that will be introduced.

Indeed, one interesting consequence of this synthetic approach is that it generates results that are important not because they raise questions about empirical results, but rather because they raise questions about the similarities between one type of model (artificial neural networks) and another (the model of associative learning proposed by Rescorla and Wagner, 1972).

1.5 Purposes and Intended Readership

1.5.1 Purposes

In providing a study of the relationship between perceptron learning and the Rescorla-Wagner model, this monograph has several purposes.

First, it provides an introduction to a particular artificial neural network with a long history, the perceptron (Rosenblatt, 1962). It also provides an introduction to modern variants of this network created when activation functions are modified.

Second, this book provides a detailed tutorial showing how perceptrons can be used to simulate a variety of standard paradigms in the animal learning literature. This includes the creation of training sets, the actual training of networks, and the interpretation of the internal structure of a network at the end of training to inform associative theories of learning. The software used to conduct the simulations reported below is available as freeware from the author's website (*http://www.bcp.psych.ualberta.ca/~mike/Software/Rosenblatt/index.html*), and it is hoped that the reader will be motivated to use this software to perform their own explorations of associative learning.

Third, this book provides a case study of the relevance of associative learning to foundational issues in cognitive science. As was noted in the previous section, the tri-level hypothesis is a fundamental property of cognitive science. Unfortunately, other common characteristics of cognitive science include intense criticism of associationism (e.g. Bever, Fodor & Garrett, 1968) and of artificial neural networks (e.g. Fodor & Pylyshyn, 1988). However, the tri-level hypothesis can be applied to both associationism (Shanks, 1995) and to connectionism (Dawson, 1998). One difficulty students have with distinguishing the computational level from the algorithmic level in theories of cognitive science is the lack of concrete examples of this distinction, as well as of its importance. This manuscript provides a detailed analysis that leans heavily upon this distinction, and which serves as a providing a needed case study.

1.5.2 Intended Readership

With these purposes in mind, a number of different types of readers will find this manuscript of interest.

First, it is primarily intended to be read by senior undergraduate students and by graduate students who may have little technical experience with either artificial neural networks or with models of associative learning. At the end of this monograph these readers will be well versed in a particular type of connectionist model and with its relationship to an important theory of associative learning.

Second, it is hoped that readers with a strong background in one of the main topics of the book, but not the other, will also benefit. In spite of a long history relating connectionism to associative learning, research in these two domains is becoming increasingly specialized, and the relationship between them is frequently underutilized or ignored. Hopefully this second type of reader will realize this after reading this manuscript, and will be inspired to conduct research that is more integrative in nature.

Third, readers with a strong background in both of the main topics of this book will also find some technical novelties in it. For example, one important issue that arises is the relationship between the threshold of an activation function in a perceptron and background or context in a conditioning experiment. The implications of modifying activation functions are also central. The formal analysis relating the Rescorla-Wagner model to perceptrons with nonlinear activation functions is new. Finally, this manuscript raises important issues concerning the need for making explicit statements about how the associative strengths modified by the Rescorla-Wagner model should be converted into behavior or responses.

1.6 What Is This Book Not About?

Before beginning, it is important to alert the reader to topics that are not central to this book. For insight into these topics, the reader will have to look elsewhere.

1.6.1 Multi-layer Perceptrons

One of the main reasons for the resurgence of interest in artificial neural networks was the discovery of rules that could be used to train multi-layer perceptrons. A multi-layer perceptron is a network that, like a perceptron, has input units and output units. However, in addition it has at least one layer of hidden units that lay between the input and output layers. Hidden units can be viewed as devices that detect important, nonlinear features in input stimuli, and they permit networks to be extremely powerful information processors.

This book is not about multi-layer perceptrons. It is exclusively about simpler networks that are largely ignored in the modern connectionist literature, as well as in modern treatments of associative learning. Fortunately, if the reader is interested in learning the technical details about such networks, there are a number of excellent resources available (De Wilde, 1997; Freeman, 1994; Gluck & Myers, 2001; Kasabov, 1996; Pao, 1989; Ripley, 1996; Rojas, 1996). Indeed, a general treatment of how these networks relate to animal learning has recently appeared (Enquist & Ghirlanda, 2005).

Of course, one reason that this book is not about multi-layer networks is because it shows that simpler networks are far more interesting that one might expect. For example, standard limitations of perceptrons can be overcome by altering their activation functions. As well, the responses of perceptrons trained in animal learning paradigms reveal surprising information that is important to consider when theorizing about associative learning.

1.6.2 Temporal Models

Any introductory book on animal learning reveals that temporal factors are crucial determinants of classical conditioning. Such factors are not treated in this book, apart from the fact that when networks are trained, they are trained by presenting one stimulus at a time. The reason that they are not treated is because temporal variables (e.g., timing of presentation of the US) are neither part of perceptrons nor part of the Rescorla-Wagner model, which are the two theories of main interest in this book. Fortunately, there exist excellent treatments of neural networks that explicitly incorporate such variables when modeling association (Schmajuk, 1997), and theoretical treatments of the importance of considering such variables (Gallistel, 1990; Gallistel & Gibbon, 2000).

While the current book does not provide any models of temporal factors, it is hoped that some of the themes that it develops concerning association and behavior, and levels of analysis, will also be applicable to more ambitious models of learning.

1.6.3 Rescorla-Wagner Alternatives

The Rescorla-Wagner model was influential because of its simplicity, and because of its ability to account for a diversity of results. However, there are many phenomena that it fails to explain (Miller et al., 1995). As a result, there have been a number of attempts to update it, or to provide alternatives to it (Krushke, 2001, 2003; Mackintosh, 1975; Pearce, 2002; Pearce & Hall, 1980; Rescorla, 2003; Wagner, 2003).

The present book focuses its treatment on the standard version of the Rescorla-Wagner, and does not attempt to expand, elaborate, or extend this model. However, it does raise the possibility that perceptrons can offer solutions to some of the Rescorla-Wagner model's problems. In particular, it explores the idea that variations of the activation function used in perceptrons can be used as different theories about how associative strengths are converted into responses.

1.7 What Is This Book About?

1.7.1 What Lies Ahead

The purpose of this book is to explore the relationship between a particular artificial neural network, the perceptron, and a particular theory of animal learning, the Rescorla-Wagner model. This exploration proceeds as follows:

Chapter 2 introduces the traditional notion of the perceptron. It does so by describing how the information processing of a neuron can be functionally described, and how such "digital neurons" can be modified by either the Hebb rule or the delta rule. Chapter 3 translates this introduction directly into the animal learning literature, by providing a number of examples of perceptrons being trained in simulations of animal learning experiments.

Chapter 4 provides an account of a more modern version of the perceptron, called the integration device. The integration device replaces the discrete step function of the traditional perceptron with its continuous approximation, the logistic equation. A number of additional simulations of animal learning experiments are then presented with this architecture. The empirical results of Chapters 3 and 4 provide evidence for a strong relationship between network and animal learning.

Chapter 5 examines this relationship at a formal level. It introduces the Rescorla-Wagner model, and provides a version of the existing linear proof of the relationship between it and a "special case" of the delta rule. It then augments this approach by showing that the learning rule for a perceptron that uses a nonlinear activation function can be translated into the Rescorla-Wagner equation.

Given the empirical and formal relationships that are established in the early chapters, a number of design decisions must be kept in mind when animal learning paradigms are simulated. These design decisions are discussed in Chapter 6. Three such decisions are explored: the relationship of a network's bias to the influence of uncontrolled background stimuli, the empirical and logical need to include a null condition, and the ability to replace a monotonic activation function with a nonmonotonic activation function. Importantly, these design decisions affect the responses of a network when it is trained in a simulated experiment.

Chapter 7 presents the critical empirical result of this book: in spite of the formal equivalence between the delta rule and the Rescorla-Wagner model, these two approaches to learning actually make different predictions. This chapter presents several examples of perceptrons responding in ways that the Rescorla-Wagner model does not predict.

Chapter 8 provides an analysis of how the "perceptron paradox" of Chapter 7 emerges. It shows that while the two approaches are equivalent at the computational level, they represent very different algorithms for learning. In one, learning occurs only after an agent responds. In the other, associative strength can be modified without the agent responding at all. The implications of this analysis to the study of associative learning are discussed in Chapter 9.

1.7.2. What Lay Behind

This manuscript would not be possible without kind support from many different quarters. The research reported here was funded by research grants from both NSERC and SSHRC. Two anonymous reviewers, several colleagues, students, and former students inspired me to think about these issues, provided me feedback on draft versions of the manuscript, or offered a forum for me to try to keep my thinking less muddled. These people include Leanne Willson, Vanessa Yaremchuk, Chris Sturdy, and Marcia Spetch. Finally, I would like to thank my wife Nancy Digdon and my children Michele and Chris for their continual support.

Chapter 2: The Perceptron

The purpose of this chapter is to introduce the general properties of a particular artificial neural network, the perceptron. We begin by considering it as a functional description of a neuron. We then consider how the perceptron can be related to associative learning in general, and to classical conditioning in particular. The chapter ends by considering two different mathematical learning rules that can be used to modify the associations in the perceptron. The second of these, the delta rule, will later play a key role in relating machine learning (i.e., the perceptron) to animal learning.

A peer-reviewed monograph published by Comparative Cognition and Behavior Reviews on behalf of the Comparative Cognition Society.

2.1 Neuronal Inspiration

2.1.1 Functional Explanations

Functional explanations are one of the most popular and effective tools adopted by cognitive scientists (Cummins, 1983; Fodor, 1975). They are used to explain the behavior of a system by appealing to the function or role of each of the system's components, and by noting how these components are organized. For example, a flow chart illustration of a computer algorithm is a functional account of how that algorithm works.

Functional explanations "abstract over" the physical properties of components. They are concerned with what a component does (i.e., its role or function), and are not concerned with what a component is made of. Thus in a functional account of the circulatory system, the role of "pump" can be served by a biological muscle (the heart) or by a pump constructed from plastics and electronics (the Jarvik-7 artificial heart). Though these two pumps are constructed from dramatically different substances, they can be viewed as being functionally equivalent, because they serve the same role.

In cognitive science, cognition is assumed to be information processing (Dawson, 1998). There is a one-to-many relationship between a particular instance of information processing and the many different physical platforms that could carry this information processing out (e.g., Fodor, 1975). As a result, functional explanations are preferred by cognitive scientists because they focus on regularities common to potentially many different physical systems (Arbib, 1964). Functional explanations are powerful, abstract accounts that are simplifications in the sense that they hide potentially complicated and messy physical details.

2.1.2 Neuronal Function

An example of an extremely complicated physical system is an individual neuron (Levitan & Kaczmarek, 1991). However, a simple functional account of a neuron can be used to describe it as an information processor.

An individual neuron receives input signals through a tree-like structure of dendrites. An individual dendrite can be thought of as a transmission line through which a weak electrical signal, called a graded potential, travels. The magnitude of a graded potential, and the distance that it travels, is directly related to the strength of the synaptic event that caused it. In other words, the greater the initiating signal, the stronger is the graded potential.

Graded potentials are usually very low amplitude electrical events. As a result, a single graded potential is not likely to have much of an effect. However, graded potentials from a number of different dendrites in a neuron's dendritic tree eventually can be summed together to produce an overall or "net" electrical effect at the neuron's soma. So, after inputs are received, a neuron can be viewed as a device the computes this net electrical effect by adding a number of different graded potentials together.

If the net effect of the graded potentials is sufficiently extreme, then the neuron will generate an action potential that will serve as an output signal to be sent to other neurons. Once this occurs, the signal does not reflect the size of the event that caused it to be initiated – the action potential is of a set, maximum amplitude. This is called the all-or-none law (e.g., Levitan & Kaczmarek, 1991, p. 41). The only way that a neuron can output information about stimulus intensity is by varying the frequency of action potentials of constant amplitude.

Functionally speaking, then, neurons receive signals via graded potentials, sum these signals together, and depending on this sum generate a response that can be sent to other neurons. Artificial neural networks are simple devices that simulate this functional account of a neuron.

2.2 A Digital Output Unit

2.2.1 Perceptron as Neuron

Frank Rosenblatt developed one of the most famous and historically important artificial neural networks, the *perceptron* (Rosenblatt, 1962). In general, a perceptron consists of one or more input units that are connected to one or more output units. The input units are used to encode a stimulus, and to then send a signal about a stimulus through the connections to the output units. The output units process the incoming signal and generate a response. In short, the output units represent the perceptron's response to the stimulus that was presented to it.

Let us assume that input units are passive – that is, that their only role is to faithfully encode an environmental stimulus, and to pass it along to the output units. Under this assumption, if a perceptron has any interesting properties, then these must depend on the kind of "processing" that the output units perform. To understand this processing, we will consider the operation of a single output unit. It will become apparent that this output unit is essentially a simple, functional, and formal account of a single neuron.

2.2.2 Computing Net Input

We saw earlier that whether or not a neuron generated an action potential depended upon the net effect of many graded potentials that were combined at the soma. In a similar fashion, the response of a single output unit depends on the net effect of many different signals being sent to it from any input units to which it is connected. The first processing step for the output unit is to compute this net effect.

A perceptron is a formal system which can be described in purely mathematical terms. In this formalism, all of the signals that are being sent from the input units to an output unit are numbers. As a result, the net effect of these signals can be determined by adding all of these numbers up into a single sum. We will call this sum a unit's *net input*.

2.2.3 Converting Net Input to Activity

In a neuron, the analog net effect of the graded potentials is converted into a digital response consistent with the "all or none" law. That is, a net effect below a critical value results in the neuron not responding at all. A net effect above a critical value results in the neuron generating an action potential. The magnitude of the action potential, if generated, is independent of the magnitude of the net effect that triggered it. A net effect slightly higher than the critical value and a net effect twice as high as the critical value both produce an action potential of the same magnitude. In short, functionally speaking a neuron is digital: it converts a wide range of analog net effects into one of two states, "On" or "Off".

Rosenblatt (1962) made the output unit of the perceptron digital in exactly this same way. The second processing step carried out by an output unit is to convert its (analog) net input into a digital on or off response.

This is accomplished using an *activation function*. Rosenblatt (1962) assumed that each output unit had a critical threshold value, represented as θ. The activation function determined an internal level of activity for the output unit by comparing the net input, net, to this threshold. If the net input exceeded the threshold, then the internal activity of the output unit was "on" (e.g., 1). Otherwise, the internal activity of the output unit was "off" (e.g., 0).

More formally, Rosenblatt (1962) used the Heaviside step function *H(net)* to convert net input into internal activity:

$$H(net) = \begin{cases} 1 : net > \theta \\ 0 : net \leq \theta \end{cases} \quad (1)$$

2.3 Activation and Response

2.3.1 What Is Activity?

In Rosenblatt's (1962) perceptron, the activation function converts an output unit's net input into the unit's internal activity. This internal activity can be considered as the response of the output unit to the stimulus that was presented to the perceptron's input units.

Rosenblatt (1962) was not the first to formalize the "all or none" law of the neuron by employing the Heaviside step function as an activation function. For example, McCulloch and Pitts (1943) adopted this strategy in their formalization of neural mechanisms. The operations of what is now known as the McCulloch-Pitts neuron are equivalent to those of an output unit in a Rosenblatt perceptron.

McCulloch and Pitts (1943) formalized neural processing in this fashion because one of their goals was to attach an interpretation to the activity of their processing units. In particular, they conceived their artificial neurons as devices for computing logical propositions. "Many years ago one of us … was led to conceive of the response of any neuron as factually equivalent to a proposition which proposed its adequate stimulus" (McCulloch & Pitts, 1943, p.21).

That is, the stimuli presented to the inputs of a McCulloch-Pitts neuron indicated the values of certain variables. The purpose of the neuron was to judge the relationship between those values. If the neuron turned "On", then this indicated that the predicate being computed by the neuron was "True" for the presented values. If the neuron was "Off", then this indicated that the predicate was "False".

For example, consider a McCulloch-Pitts neuron that receives two different incoming signals. The signals indicate whether two variables, X and Y, are either present or absent. Imagine that the response of the neuron is as described in Table 2-1. In this situation, the neuron is computing the logical operator AND over the two variables.

Variable X	Variable Y	Response
Absent	Absent	False
Absent	Present	False
Present	Absent	False
Present	Present	True

Table 2-1. Responses of a McCulloch-Pitts neuron to two variables when the neuron is computing AND.

We are not required to interpret a perceptron's output unit as being a logical device. More generally, "perceptrons make decisions – determining whether or not an event fits a certain 'pattern' – by adding up evidence obtained from many small experiments" (Minsky & Papert, 1988, p. 4). Perceptrons are particularly well-suited for assigning discrete responses to noisy, analog stimuli. For this reason, perceptrons are often described as prototypical machines for pattern recognition or pattern classification.

While it is not necessary to do so, the patterns that perceptrons can recognize or classify are often assumed to be visual or spatial. The response of a perceptron is often interpreted as indicating whether a particular geometric property is true of a visual stimulus (Minsky & Papert, 1988) or as indicating the name of the class to which the visual stimulus is assigned.

2.3.2 Activity and Response

We can easily interpret a perceptron as a pattern classifier or as a logical operator if we assume that the activation function represents the perceptron's response. This in turn could signify an observable behavior, or an internally generated expectancy (e.g., probability of being reinforced). We will also see in later chapters that the range of responses that a perceptron generates to a set of stimuli can be affected by changing the activation function. That is, there are a number of different "flavors" of perceptron, with each different flavor using a different activation function. The kinds of judgments that a perceptron makes depend crucially upon the properties of the particular activation function that it employs.

2.4 Association in the Perceptron

2.4.1 Stimulus and Response

Classical conditioning, as originally conceived (Pavlov, 1927), is a form of learning that is based on the associationist law of contiguity (Warren, 1921). According to the law of contiguity, when two ideas (e.g., A and B) generally occur together then the occurrence of one of the ideas alone can then lead to the occurrence of the other.

The law of contiguity was used by the associationists to account for the sequential production of thoughts. "Ideas that are themselves are not at all of kin, come to be so united in some men's minds that it is very hard to separate them, they always keep in company, and the one no sooner at any time comes into the understanding but its associate appears with it" (Locke, 1977, p. 219).

One way to consider this sequential production of thoughts is to view two associated ideas as standing in a stimulus-response relationship. For example, at one point in time idea A is present for some reason. This idea serves as a stimulus which, via association, brings idea B to mind. Idea B can be described as the response to the stimulus (i.e., to idea A).

Pavlov's (1927) study of classical conditioning required some stimulus-response relationships to be present prior to the learning of new associations. In particular, some stimulus (called the unconditional stimulus, or the UCS) would already produce a behavioral response (called the unconditional response, or the UCR). For example, in his famous experiments on conditional learning in dogs the presence of food was a UCS that caused the dogs to salivate (the UCR). This stimulus-response relationship was established before Pavlov's conditioning experiments began.

2.4.2 UCS→UCR in the Perceptron

A very simple form of the perceptron can be used to implement the relationship between a UCS and a UCR that was required by Pavlov (1927). Let us begin with a perceptron that has a single input unit and a single output unit. The input unit will be used to represent the presence of the UCS – when the UCS is present, the input unit will be activated with the value of 1, and when it is absent the input unit will be activated with the value of 0. Our desire is to have the output unit turn "On" when the input unit is active, and to turn "Off" when the input unit is not active.

Our first step in achieving this goal will be to assign a threshold value θ to be used in the Heaviside activation function of the output unit. Let us assume that $\theta = 0.5$.

Our second step in achieving this goal will be to assign a connection between the input unit and the output unit. This connection is a conduit through which the input unit will send a signal to the output unit. The connection has assigned to it a numerical value called its weight (*w*). The weight of a connection is used to scale the numerical signal being sent through it. Let an input unit at one end of the connection send the activation value *a*. The signal that comes out of the connection at the other end, and which is delivered to the output unit, is equal to the product of the connection weight and the input activity (i.e., *wa*).

Our third step in achieving the goal of getting the output unit to activate when the input unit activates is to assign a value to the connection weight. If we let $w = 1$, our perceptron will be complete. When $a = 0$, $wa = 0$ which is less than θ. As a result, the activation function in the output unit will generate a value of 0 when the UCS is absent. In contrast, when $a = 1$, $wa = 1$ which is greater than θ. As a result, the activation function in the output unit will generate a value of 1 when the UCS is present.

2.5 Multiple Associations

2.5.1 Multiple Inputs and Outputs

The preceding perceptron was particularly simple because it had only one input unit and only one weighted connection. In most cases, a perceptron will have more than one input unit, and in many cases it will have more than one output unit. As a result, it will have multiple connections. We need to generalize our formal description of the perceptron to take these conditions into account.

Consider a perceptron that has *N* different input units. One of these input units could be identified as input unit *i*. The activation of this input unit would be represented as the value a_i. Imagine that this perceptron also has *M* different output units. One of these output units could be identified as output unit *j*. The net input for this output unit would be represented as the value net_j, and the activation for this output unit would be represented as the value a_j. The threshold for this output unit would be represented as the value θ_j.

A perceptron with *N* different input units and *M* different output units will have *N* x *M* different connections, assuming that the network is fully connected (i.e., assuming that every input unit is connected once to every output unit). Each of these connections might have a different weight. We will represent the weight of the connection from input unit *i* to output unit *j* as the value w_{ij}.

Each output unit *j* must compute its net input, which is the sum of the signals from each input unit *i*. Each signal from an input unit is a value a_i times a value w_{ij}. The net input of output unit *j* must therefore be:

$$net_j = \sum_{i=1}^{M} a_i w_{ij} \qquad (2)$$

In a traditional perceptron, after computing its net input, output unit j will compute its activation a_j using the Heaviside equation. Formally this can be expressed as:

$$a_j = H(net_j) = \begin{cases} 1 : net_j > \theta_j \\ 0 : net_j \leq \theta_j \end{cases} \qquad (3)$$

2.5.2 An Example Perceptron

Previously we described a McCulloch-Pitts neuron that computed the AND of variables X and Y. Let us describe a functionally equivalent perceptron.

For any input unit *i*, let a_i be equal to 0 if the variable represented by the input unit is absent, and let a_i be equal to 1 if the variable is present. Let input unit 1 represent the status of variable X, and let input unit 2 represent the status of variable Y. Assume that the perceptron has one output unit, unit *j*. The threshold of this output unit, θ_j, has the value of 1.5. There are two connections in this perceptron. The first is the connection from input unit 1 to output unit j, which has the connection weight w_{1j}. Let w_{1j} be equal to 1. The second is the connection from input unit 2 to output unit j, which has the connection weight w_{2j}. Let w_{2j} also be 1.

The perceptron is now complete. There are four possible combinations of variables X and Y that can be presented to it; each is described as a row in Table 2-2. For each, the output unit will compute its net input and its activity according to the two equations above. The results of these calculations are also provided in Table 2-2. Note that the activity of output unit j corresponds to the behavior of the McCulloch-Pitts neuron described in Table 2-1. In other words, we have designed a perceptron for computing AND.

a_1	a_2	net_j	a_j
0	0	0	0
0	1	1	0
1	0	1	0
1	1	2	1

Table 2-2. Responses of a perceptron that ANDS two inputs.

2.6 Learning in Perceptrons

2.6.1 Neurons That Learn

In terms of its structure, and its basic mechanisms of operation, Rosenblatt's (1962) perceptron is functionally equivalent to a McCulloch-Pitts neuron. However, there is a fundamental difference between McCulloch-Pitts neurons and perceptrons. In order to design a network of McCulloch-Pitts neurons to perform a particular task, one had to "hand wire" or pre-set all of the threshold values, and all of the connection weights, in the entire network. In contrast, perceptrons are generally not hand wired. Rosenblatt developed a procedure in which the structure of a perceptron was acquired through experience. Perceptrons are functionally equivalent to neurons that learn.

Perceptron learning involves two general components. The first is a training regimen in which a set of training stimuli – and an associated set of desired responses – is presented to the perceptron. The second is a learning procedure that uses the responses of the perceptron to modify the connection weights and the thresholds of the perceptron. These modifications are performed in such a way that over time (i.e., over a number of stimulus-response pairings) the structure of the perceptron changes in such a way that a desired set of stimulus-response pairings is achieved.

Recall that one interpretation of a connection weight was that it represented the strength of association between a stimulus and a response. Perceptrons are used to model associative learning in the sense that their connection weights (associative strengths) are modified via presentations of stimuli (experience).

2.6.2 Supervised Learning

In order to teach a perceptron to perform a desired set of stimulus-response pairings, supervised learning must be employed. Supervised learning is a situation in which there are two agents – one that learns, and another that teaches. The learning agent experiences its environment and adapts to it, using basic learning mechanisms. The teaching agent – the supervisor – manipulates the learning agent's environment in such a way that the latter adapts to the environment in a particular way.

The supervised training of a perceptron is analogous to classical conditioning studied in experiments on animal learning (Pearce, 1997). In these experiments, the animal serves the role of the learning agent, and brings to this role its intrinsic learning mechanisms. The experimenter serves the role of the supervisor, and manipulates the animal's environment in such a way that a desired pattern of learning – that is, desired by the experimenter – is observed.

The defining characteristic of supervised learning is that the supervisor manipulates the learning agent's environment to produce desired results. For example, classical conditioning can be viewed as supervised learning in the sense that the supervisor determines whether the US is presented on any given trial.

Some researchers have argued that the need for supervised learning to have an external teacher with knowledge of desired outcomes makes this type of learning biologically implausible (Parisi, Cecconi, & Nolfi, 1990). In nature, animals do not have access to such teachers, and must therefore learn in other ways. As a result, many proposals for unsupervised learning exist in which systems simply acquire the statistical regularities in their environment (Grossberg, 1988; Kohonen, 1977, 1984).

However, if supervised learning is merely viewed as being analogous to an experimental situation in which a learning agent's environment is being manipulated, and the agent is responding in a natural way to these manipulations, then it is a perfectly plausible approach. Indeed, we will attempt to recast the notion of supervision into the less controversial notion of reward or punishment. For this reason, we will be exclusively interested in examining supervised learning situations.

2.7 Hebb Learning

2.7.1 The Law of Exercise

According to the associative principle of contiguity, if two ideas occur near one another in space or in time, then the association between these two ideas should become stronger.

An example of the principle of contiguity in the study of animal learning is the law of exercise (Thorndike, 1932) which "asserts that, other things being equal, the oftener a situation connects with or evokes or leads to or is followed by a certain response, the stronger becomes the tendency for it to do so in the future" (p. 6). The law of exercise is an example of the principle of contiguity being applied to the association between a stimulus and a response.

Since the work of (James, 1890), researchers have investigated a neural basis of the law of exercise. One famous proposal was provided by Hebb (1949, p. 62): "When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased." This idea still inspires a great deal of modern research. For instance, the December 2002 issue of the journal *Biological Cybernetics* is exclusively devoted to Hebb-style learning.

2.7.2 The Hebb Rule

With artificial neural networks like perceptrons, Hebb learning amounts to applying a multiplicative rule (Dawson, 2004, Chapter 9). Given a connection between an input unit *i* and an output unit *j*, the change in connection weight w_{ij} is defined as the multiplication of the activities of the two units, as well as of a fractional learning rate η which regulates the speed of learning:

$$\Delta w_{ij} = \eta \cdot a_i \cdot a_j \qquad (4)$$

The issue is how to cast this rule, called the Hebb rule, into the context of supervised learning. One approach could be to exploit the required relationship in classical conditioning between the US and the UR. That is, one could present the US to automatically activate the output unit of the perceptron. At the same time, one could activate an input unit representing the presence of a CS. The Hebb rule could then be applied to modify the connection between the activated input unit and the activated output unit. This approach is illustrated in Figure 2-1.

In Figure 2-1, an output unit is connected to two input units that can represent the presence of two CSs, A and B. In the illustrated case, the experimenter has presented CS_A, and at the same time presents the US to activate the output unit. According to the Hebb rule, the connection weight between the two units (w_{Aj}) would be increased. The other connection weight would not be increased, because CS_B was not presented at the same time that the output unit was activated by the US.

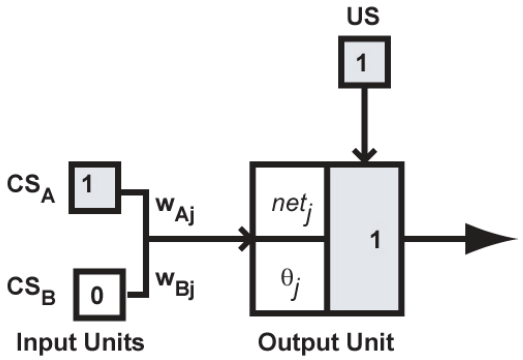

Figure 2-1.
A perceptron that learns according to the Hebb rule. In this example, only w_{Aj} will change.

2.8 Pros and Cons of Hebb Learning

2.8.1 Hebb Rule Advantages

The Hebb rule has been actively studied for over half a century because it offers a number of advantages. First, it is consistent with one of the most enduring associative laws, the principle of contiguity. Second, it is a simple and elegant law. Third, it generates some results that are nicely consistent with studies of associative memory in humans and animals (Eich, 1982; Gluck & Bower, 1988; Hinton & Anderson, 1981; Murdock, 1982, 1997). Fourth, the Hebb rule seems to be in agreement with biochemical accounts of learning in the hippocampus, in particular the type of learning associated with NMDA receptors (Brown, 1990; Cotman, Monaghan, & Ganong, 1988; Gluck & Myers, 2001; Martinez & Derrick, 1996; Martinez & Kesner, 1998).

2.8.2 Hebb Rule Disadvantages

In spite of these many advantages, the Hebb rule is not the dominant learning rule studied by artificial neural network researchers. This is because the Hebb rule also has a number of serious disadvantages.

First, in its simplest state the Hebb rule is such that associations can grow to an unlimited size. As a result, in a system that is governed by the version of the Hebb rule that we have seen, eventually everything becomes associated with everything else, and discrimination between stimuli is impossible. One solution to this problem is to include some form of inhibition (Milner, 1957; Rochester, Holland, Haibt, & Duda, 1956). Another solution is to include additional mechanisms that result in connection weights decaying over time (Gerstner & Kistler, 2002).

Second, a system governed by the Hebb rule is easily disrupted by noise (Dawson, 2004; Jordan, 1986). For instance, for perfect learning all stimuli and responses have to be completed uncorrelated. As soon as some correlation appears between stimuli, errors in responses will be observed.

Third, the Hebb rule does not obey the sensible rule "if it ain't broke, don't fix it." That is, even if the existing associative strength between an input unit and an output unit is sufficient for the network to respond correctly, the Hebb rule will unnecessarily change the connection weight between the units (Dawson, 2005). This is one reason that all of the connection weights can grow without bound.

2.8.3 Learning from Mistakes

One approach to solving all of these problems with the Hebb rule is to create a variation of the rule that is sensitive to error. If the network does not make an error, then its connection weights are untouched by a more advanced learning rule. If an error is made, then connection weights are modified in such a way that the error is decreased.

In general, when learning from mistakes, the network takes an input stimulus and responds to it before any learning takes place. This observed response is then compared to a desired response, which is used to create an error signal. The error is then used to modify the connection weights.

This type of supervised learning – learning from mistakes – is almost always used to train perceptrons because of the limitations of the Hebb rule. As a result, the type of simple associative learning that was illustrated in Figure 2-1 (see also Gluck & Bower, 1988, Figure 1; Pearce, 1997, Figure 5-18) is generally not true for the perceptron. A more complicated relationship between the US and the output unit's response is required to indicate how a perceptron can learn from its mistakes.

2.9 The Delta Rule

2.9.1 Defining Error

One of Rosenblatt's (1962) many contributions was his statement of an "error-corrective reinforcement system" that could be used to train simple neural networks (p. 91, Definition 41). His system was a training procedure that only changed the weights of a perceptron when it generated an incorrect response. A similar rule, the Widrow-Hoff rule, was developed independently at roughly the same time (Widrow & Hoff, 1960). Most modern connectionist researchers call this type of error correcting rule the *delta rule* (e.g., Gluck & Myers, 2001; Quinlan, 1991; Schmajuk, 1997; Stone, 1986).

The delta rule is a variation of Hebb learning that can be used to teach a perceptron from its mistakes. In this rule, the difference between the observed response of a perceptron and the desired response is used to modify connection weights.

For this rule, it is important to formally define an output unit's error. Let us assume that the actual response of output unit *j* is its activity a_j. Let us also assume that an external teacher has provided a target or desired value for output unit *j*, which we can designate as t_j. The output unit's error, δ_j, is the difference between the target and observed values:

$$\delta_j = t_j - a_j \qquad (5)$$

2.9.2 Delta Learning Of Associations

In the Hebb rule, the change in the association between an input unit and an output unit was defined as the triple product of a learning rate, the activity of the input unit, and the activity of the output unit. The delta rule is very similar in structure. The only difference between it and the Hebb rule is that the activity of the output unit is replaced with the output unit's error:

$$\begin{aligned} \Delta w_{ij} &= \eta \cdot a_i \cdot \delta_j \\ &= \eta \cdot a_i \cdot (t_j - a_j) \end{aligned} \qquad (6)$$

The logic of this rule is that the error value will result in a change in weight that will move the net input to the output unit in the correct direction to generate the desired output (see Dawson, 2004, p. 163). If error is equal to zero, then no change is required, and the equation above will produce a Δw_{ij} of 0. If the net input is too large, the error will be negative, and will produce a negative Δw_{ij} that decreases the connection weight (and therefore decreases the net input the next time that stimulus occurs). If the net input is too small, the error will be positive, and will produce a positive Δw_{ij} that increases the connection weight (and therefore increases the net input the next time that stimulus occurs).

2.9.3 Supervised Learning and US

The delta rule requires us to think about an experimenter's control of learning in a slightly more complicated way. The US is not used to activate the perceptron's output unit. Instead, it is used to provide information to a "comparison mechanism" that determines what the target value for the output unit is. In the simplest case, the US is equal to t_j. Therefore if the comparison mechanism receives both the US and the perceptron's activation value then it can compute the error term , and then use this error term to modify connection weights. This is illustrated in Figure 2-2, which is intended as an elaboration of Figure 2-1.

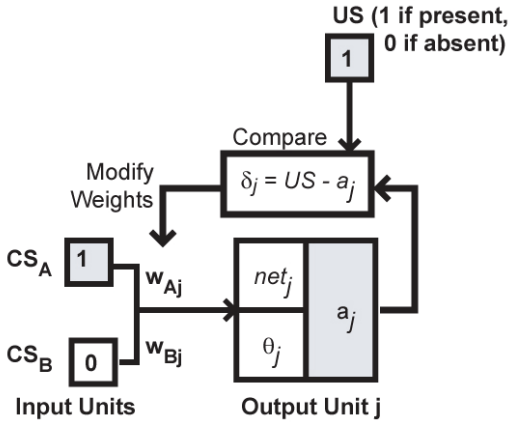

Figure 2-2.
A perceptron that learns according to the Delta rule.

Chapter 3: Associative Learning in the Perceptron: Case Studies

The purpose of this chapter is to empirically study the perceptron from Chapter 2 in the context of associative learning. After introducing some of the general characteristics of the simulation methodology that was used, perceptrons are trained in a small number of classical paradigms from the associative learning literature. In each case, a phenomenon that has been observed in animal learners is replicated in the computer simulation. This provides initial empirical support for the claim that perceptron learning is strongly related to at least some forms of animal learning.

A peer-reviewed monograph published by Comparative Cognition and Behavior Reviews on behalf of the Comparative Cognition Society.

3.1 Studying Associative Learning in the Perceptron

3.1.1 Setting the Stage

We have developed the learning rule for the perceptron in the context of the study of associative learning in animals. For example, we have included information about the CS and the US in Figures 2-1 and 2-2. Our goal is to move towards a more formal treatment of the relationship between perceptrons and traditional approaches to learning.

Our next step towards achieving this goal is to take the formal machinery that has already been discussed and demonstrate it in action. To be more precise, we will examine the results of a computer simulation that will be used to train perceptrons in a small number of classic animal learning paradigms. We will empirically demonstrate a relationship between perceptrons and models of animal learning by observing correspondences between the simulation results and those obtained from studies of animals.

3.1.2 Technical Details

All of the simulations that are described in the following chapters were conducted with a program called Rosenblatt (Dawson, 2005). The Rosenblatt program is available as freeware from: http://www.bcp.psych.ualberta.ca/~mike/Software/Rosenblatt/.

Each simulation trains a perceptron that has two input units, each used to represent the presence or absence of a conditioned stimulus (CS_A or CS_B). It also has one output unit that is used to generate a response to the stimuli that are presented to the perceptron.

The output unit uses a step activation function to compute its activity value a_j. First, it computes its net input, net_j, which is the sum of the weighted signals that are being received from the input units. Second, it adds its threshold θ_j to net_j. If the result is greater than 0, then the value of a_j is 1. Otherwise the value of a_j is 0. This is the same as determining whether or not net_j is greater than θ_j.

The perceptrons are trained with Rosenblatt's (1962) delta rule. The learning rate η is equal to 0.5 (unless otherwise noted). The networks are taught with a "sweep-wise" procedure. In a single sweep or epoch, every training pattern is presented once. However, within a sweep the presentation order of the training patterns is random. Every training pattern is associated with a desired response for computing the error for modifying the connection weights as illustrated in Figure 2-2.

The perceptron's weights are updated after every presentation of a training pattern. The connection weights are changed using the delta rule. The delta rule is also used to change the threshold of the output unit. This is accomplished by assuming that θ is the weight of an additional connection between the output unit and an input unit that is always on (i.e., that always has a value of 0). Under this assumption, a unit's threshold can easily be trained (e.g., Dawson, 2004, p. 181).

Training ends once the network has converged to a solution to the presented problem. That is, training comes to a halt when the network generates a "hit" to every pattern. A "hit" occurs when the perceptron turns on to every pattern that has a target response of 1, and when it turns off to every pattern that has a target response of 0. In short, when the output unit's error is totaled over every pattern presented in a sweep, and equals 0, then the perceptron has learned the desired task and training can stop.

Before training, every connection weight is randomly assigned a value from the continuous range [-0.1, 0.1]. The initial value of the threshold is 0. At the end of training, the connection weights and the threshold will have adopted specific values that mediate particular responses to particular stimuli. It is often interesting to examine these values at the end of training (e.g., Dawson, 2004).

3.2 Simple Association

3.2.1 Classical Conditioning

Pavlov's (1927) fundamental discovery was the conditioned reflex. Simply stated, prior to conditioning a CS is highly unlikely to produce a particular response, while a US is. However, after a series of trials in which the US and the CS are presented together, the CS will produce the response.

The simulation below demonstrates this type of associative learning. Separate associations involving two CSs (CS_A and CS_B) are taught to a perceptron. Prior to training, neither CS will cause the output unit to respond. However, after a short period of training with the delta rule, either CS will cause the output unit to generate an activation value of 1. In this simulation the two CSs are also presented simultaneously, representing a compound conditioned stimulus (i.e., CS_{AB}).

3.2.2 Training Set

Four patterns are presented to a perceptron during a sweep of training. They, along with their target values t_j, are provided in Table 3-1 below. The two columns labeled CS_A and CS_B provide the activity of each input unit used to represent a CS. If a CS is present, the input unit has an activity of 1, and if it is absent it has an activity of 0. The goal of training is to produce a set of weights such that the perceptron generates a response identical to the t_j column for each training pattern.

Pattern	CS_A	CS_B	t_j
1	0	0	0
2	1	0	1
3	0	1	1
4	1	1	1

Table 3-1

3.2.3 Results

Twenty different perceptrons, each started randomly according to the procedure described in Section 3.1.2, served as "subjects". Each perceptron was trained with the delta rule, and learned to respond correctly to each pattern in the training set (that is, to generate the desired outputs (t_j) provided in Table 3-1). On average, this required 4.15 sweeps of training. The structure of a typical network after training completed is provided in Table 3-2.

Unit	Weight	θ
CS_A	0.93	-
CS_B	0.51	-
Output	-	-0.50

Table 3-2

This table reveals how either CS on its own activates the output unit after training completed. When CS_A has a value of 1, it sends a signal of 0.93 to the output unit. When combined with θ of -0.50, the net input is 0.43. This is above 0, so the output unit turns on. Similarly, when CS_B is activated, it sends a signal of 0.51 to the output unit, which will produce a net input of 0.01. Again, this causes the output unit to turn on, because it is greater than 0. When both stimuli are presented, the net input is 0.94, and again the output unit turns on.

When neither CS_A nor CS_B is present, the total signal coming to the output unit is 0. When combined with θ of -0.50, the net input is -0.50. This is below 0, so the output unit generates a 0, as desired.

The two connection weight values differ from one another because they started at different random values. If this simulation was repeated many times, and the average value of the connection weights after training was computed, it would be expected that the two average weights would be equal to one another.

Importantly, three of the different stimulus configurations (CS_A, CS_B, CS_{AB}) produce very different net inputs, but cause the same response in the output unit. This illustrates the all-or-none law: a more intense input (e.g., the compound stimulus) leads to a more intense (i.e., more positive) net input, but still causes a response of the identical magnitude as that caused by a less intense input (CS_A or CS_B alone).

3.3 Intensity of Conditioned Stimuli

3.3.1 Conditioning and CS Intensity

All things being equal, the intensity of a CS affects its ability to be classically conditioned. For example, in one study rats were conditioned by pairing the same magnitude of US with one of three different magnitudes of white noise CSs: low intensity (49 dB), medium intensity (62.5 dB) and high intensity (81 dB) (Kamin & Schaub, 1963). CS intensity was directly related to the rate of conditioning that Kamin and Schaub observed.

3.3.2 CS Intensity in the Perceptron

The intensity of the CSs used to train a perceptron can be varied by activating input units with some value other than 1. For instance, if we wanted to model CSs that were 10 times less intense than those used in the previous study, when a CS was present we would activate its input unit with a value of 0.1 instead of 1.0.

This simulation takes exactly this approach. It is exactly the same as the previous simulation, with the exception that the CSs are 10 times less intense. That is, when a CS is present, the input unit that represents it is activated with a value of 0.1 instead of the value of 1 that was used in Table 3-1. The resulting training set is given in Table 3-3; again, the point of training is to produce a set of weights such that the perceptron generates a response identical to the t_j column for each training pattern in the table.

Pattern	CS_A	CS_B	t_j
1	0	0	0
2	0.1	0	1
3	0	0.1	1
4	0.1	0.1	1

Table 3-3

20 different perceptrons were "subjects" in an experimental condition that used the Table 3-3 training set. The number of sweeps required to converge in this condition was compared to the number required by 20 additional control subjects trained with the Table 3-1 training set.

3.3.3 Results

On average, control condition perceptrons converged after 4.15 sweeps of training. If stimulus intensity affects perceptron learning in a fashion analogous to that reported by Kamin and Schaub (1963), then perceptrons in the experimental condition should take much longer to learn. Indeed, this is the case. In this condition it takes on average 197.75 sweeps to generate hits to all four training patterns. This difference is statistically significant ($t = -146.205$, $df = 38$, $p < 0.001$).

Even though decreasing stimulus values produces slower learning, the final structure of a perceptron trained in the experimental condition solves this conditioning problem in a very similar fashion to that observed in the previous simulation. Table 3-4 provides one network's values for the various network components at the end of training:

Unit	Weight	θ
CS_A	5.02	-
CS_B	5.05	-
Output	-	-0.50

Table 3-4

One of the key differences between this network and the previous network is the size of the weights. The weights in Table 3-4 are an order of magnitude larger than those in Table 3-2. However, both networks have the same threshold, so this difference makes sense. For the current network with $\theta = -0.50$, the connection weights have to be an order of magnitude larger because the stimulus intensities are an order of magnitude smaller.

This also accounts for the increased amount of stimulus presentations required for learning. Because the connection weights need to be more extreme for the weaker stimuli, more learning trials are required to make the connection weights sufficiently large for the network to generate all of the desired responses.

3.4 Intensity of Unconditioned Stimuli

3.4.1 The Effect of US Intensity

Another factor that has been shown to affect conditioning is the intensity of the US. For example, one experiment concerned conditioning eye blink responses in rat pups (Freeman, Spencer, Skelton, & Stanton, 1993). In this study, a tone CS was paired with a periocular shock US, and shock magnitude was manipulated. It was found that a 3 mA shock led to faster acquisition of conditioned eye blinks than did a 1 mA shock.

One approach to performing this type of manipulation on a perceptron would be modifying η, the learning rate. For instance, if this factor was made to be a very small fraction, then this might simulate a situation in which learning was slower because the US being used was less intense.

Take the training set from Table 3-1. In the simulation in Section 3.2, the perceptrons learned to respond to that training set after approximately 4 sweeps when η was 0.5. Let us repeat this simulation study, but reduce η to a value of 0.005 – two orders of magnitude smaller than the original study.

3.4.2 Effects of Decreasing η

When this is done in one simulation, the perceptron learns to respond to each of the three training patterns, but training is much slower. When 20 different perceptrons are trained with $\eta = 0.5$, it takes on average 4.15 sweeps to converge. In contrast, when $\eta = 0.005$ it takes an average of 12.38 sweeps to converge. This difference is statistically significant ($t = -5.288$, $df = 38$, $p < 0.001$).

Consider one of these latter networks, which converged in 21 epochs. Table 3-5 presents its structure. It learns the associations involving independently presented CSs in the same qualitative fashion as was described for Table 3-2. Quantitatively, though, the structure is markedly different. The reduction in the learning rate was also responsible for a similar reduction in the sizes of the two connection weights and the bias of the output unit.

Unit	Weight	θ
CS_A	0.007	-
CS_B	0.081	-
Output	-	-0.005

Table 3-5

3.4.2 Effects of Increasing η

Take the same training set, and train 20 more perceptrons after increasing η to 0.9.

When this is done, learning is much faster than the previous experiment: perceptrons converged on average after 4.33 epochs. However, this result is not statistically different from the networks trained with $\eta = 0.5$ ($t = -1.034$, $df = 38$, $p < 0.308$). The increase in learning rate produced a corresponding increase in the sizes of the network's connection weights, as shown in Table 3-6. Compare this structure to those presented in Tables 3-2 and 3-5, which are example perceptrons trained to solve the identical input-output problem, but were trained to do so with smaller η values.

Unit	Weight	θ
CS_A	1.78	-
CS_B	1.76	-
Output	-	-0.90

Table 3-6

Increasing η to 0.9 led to essentially the same average convergence rate than when η was 0.1. This likely reveals a floor effect in this study: when η was 0.5, learning was actually so fast that further increases in the learning rate were unlikely to speed learning up much further. If a perceptron was trained on a more difficult problem, then we would expect finer distinctions to emerge between different learning rate conditions.

3.5 Configural Representations of Compound Stimuli

3.5.1 Compound Stimuli

Pavlov (1927) reported the results of a number of experiments in which animals were conditioned to compound stimuli. A compound stimulus is created when, for instance, two different CSs are presented simultaneously. Compound stimuli are of interest because they raise questions about whether an agent creates a unique representation of the compound, or instead creates separate representations of the compound's components, whose individual effects are summed in some fashion. This issue is still one of hot debate (Wasserman & Miller, 1997). Interestingly, perceptrons can be used to study learning for both of these approaches to representing compound stimuli.

3.5.2 Elemental Representations

Some learning theorists have argued that compound stimuli are represented by activating their component elements as individual stimulus parts (Rescorla & Wagner, 1972; Wagner, 2003). According to this elemental approach, a compound stimulus does not have a unique representation as a configuration. It is instead merely represented as the collection of its parts.

Elementary representations of compound stimuli are straightforward in a perceptron. Each input unit is used to represent the presence or absence of a stimulus component. The compound stimulus is represented simply by turning the input units for each of its elements on (Gluck & Bower, 1988). This approach to representing compound stimuli was demonstrated in the simulations described in Sections 3.2., 3.3, and 3.4

3.5.2 Configural Representation

A configural representation is used to represent not only the elements of a compound stimulus, but also the unique properties of a stimulus that might emerge from its compound nature. This can be accomplished, for instance, by including extra input units in a perceptron. For example, a three-input unit perceptron could use one unit to represent the presence of CS_A, a second to represent the presence of CS_B, and a third to represent the unique attributes of the compound created when CS_A and CS_B are presented together (i.e., CS_{AB}). To represent a compound stimulus with these units, one turns the unique configuration unit on, as well as the two units representing its component stimuli. The unit for CS_{AB} is only turned on when both of the other input units are also activated. Table 3-7 presents a configural representation for another logical operator, exclusive-or (XOR).

Pattern	CS_A	CS_B	CS_{AB}	t_j
1	0	0	0	0
2	1	0	0	1
3	0	1	0	1
4	1	1	1	0

Table 3-7

Table 3-8 presents the structure of a typical network trained to solve the problem laid out in Table 3-7. Table 3-8 shows that the network's weight assignment to the two elemental components of the representation (CS_A and CS_B) is very similar to the representation of OR that we have seen in preceding sections. That is, without the CS_{AB} unit, the network would turn on when either or both elements are present. However, the configural component of the representation (CS_{AB}) presents the network from incorrectly turning on to the compound stimulus. This is because this unit has developed a strong inhibitor connection to the output unit which is sufficient to keep the output unit off when the compound stimulus is presented.

Unit	Weight	θ
CS_A	0.57	-
CS_B	0.98	-
CS_{AB}	-2.02	-
Output	-	-0.50

Table 3-8

3.6 Positive Patterning In The Perceptron

3.6.1 Positive Patterning

Compound stimuli are of interest because they provide a fascinating opportunity to explore the relationship between a whole and its parts. For example, Pavlov (1927) was the first to report that one could train an animal to respond to a compound stimulus, and at the same time train the animal not to respond when the compound's parts were presented separately. "This result is obtained by constant reinforcement of the compound stimulus, while its components, on the frequent occasions when they are applied singly, remain without reinforcement" (p. 144).

This paradigm is known as positive patterning, and is usually represented with the notation [A-, B-, AB+] (Delamater, Sosa, & Katz, 1999). The As and Bs indicate which stimuli are presented, while the +s and –s indicate whether the animal was reinforced or not. If a fourth stimulus condition is included (-) to indicate that the animal is not reinforced when no CS is present, then positive patterning is logically equivalent to the AND logical problem that was discussed in Chapter 2 (Yaremchuk et al., 2005).

3.6.2 Training Set

Table 3-9 provides the training set that can be used to teach a perceptron using the positive patterning paradigm. Note that this particular training set assumes an elemental representation of stimuli.

Pattern	CS_A	CS_B	t_j
1	0	0	0
2	1	0	0
3	0	1	0
4	1	1	1

Table 3-9

3.6.3 Results

A study was conducted in which 20 perceptrons were taught to make the responses illustrated in Table 3-9. This was an easy problem for a perceptron to solve; on average convergence was achieved after 3.96 epochs.

The structure of one network that converged after 5 epochs of training is provided in Table 3-10.

Unit	Weight	θ
CS_A	0.47	-
CS_B	0.42	-
Output	-	-0.50

Table 3-10

This table reveals that either CS on its own is not capable of activating the output unit after training is completed, because net input is negative. This is also true when both stimuli are absent. It is only when both input units are turned on that they provide a strong enough signal to overcome the negative bias and create a positive net input.

Interestingly, the structure of this network is almost identical to the structure of the network that was presented earlier in Table 3-2. That network was trained to compute logical OR, because it would turn on when A, B, or AB were presented. The only notable difference between the structures of the two networks is the relation of the connection weights to the bias: in Table 3-2, both connection weights are necessarily larger than the absolute value of the bias, while in Table 3-10, both connection weights are necessarily smaller than the absolute value of the bias.

3.6.4 The Modern Perceptron

With initial results like those that have been presented in this chapter, we will now move to the study of a more modern version of the perceptron, called the integration device. The next chapter introduces this network, describes how it can be trained, and presents a number of results that relate it to the study of animal learning.

Chapter 4: Modernizing the Perceptron: The Integration Device

The purpose of this chapter is to present a modern version of the perceptron. This is called the integration device, and is identical to the perceptron that was discussed in Chapters 2 and 3 with the exception that it uses a continuous, nonlinear activation function. This new activation function is an approximation of the discontinuous function that is described by the Heaviside equation. The chapter then turns to showing that this modern version of the perceptron is also closely related to associative learning in animals. Simulation results are provided that show how a number of standard animal learning phenomena can be replicated in an integration device. The results point to the need to determine the formal relationship between network learning and animal learning.

A peer-reviewed monograph published by Comparative Cognition and Behavior Reviews on behalf of the Comparative Cognition Society.

4.1 A Continuous Approximation of the Heaviside Equation

4.1.1 From Discrete To Continuous

The perceptrons that we have been exploring have been trained with Rosenblatt's (1962) version of the delta rule, and have used a particular discrete activation function, the Heaviside equation. The purpose of this chapter is to introduce a modern variation of this architecture. It differs from the perceptron that we have been discussing in two major respects. First, it uses a continuous activation function that approximates the Heaviside equation. Second, it uses a continuous variation of the delta rule, a gradient descent rule.

The main purpose of introducing this variant of the perceptron is to be in a position to later explore the relationship between it and two other modern developments in the study of networks. The first is the multilayer perceptron (Rumelhart et al., 1986), which has at least one layer of intermediate processing units between its input and output units. The discovery of learning rules for multilayer perceptrons depended heavily upon incorporating a continuous activation function into the processing units.

The second modern development to which this variant of the perceptron will be related is the use of alternative continuous activation functions, such as the Gaussian (Dawson & Schopflocher, 1992), in the processing units. We will see that the capabilities of a perceptron, as well as the relationship between the perceptron and models of animal learning, depend heavily upon the nature of the activation function that is chosen.

While the perceptron described in this chapter differs from the one that was presented in the previous two chapters, the two types of networks are highly related. In particular, both types of perceptrons can be trained with the delta rule. As well, the simulation results reported previously can easily be replicated in the integration device that we will now introduce.

4.1.2 The Logistic Activation Function

The main feature of the Heaviside equation is that it provides a step function that converts net input into internal activity. In it, if internal activity is below a threshold, then activity is equal to 0. If internal activity meets or exceeds a threshold, then activity is equal to 1. This function is discontinuous, because there is a "break" in it when its value goes from 0 to 1.

Modern artificial neural network researchers have replaced the Heaviside equation with a continuous function that approximates its key features. One popular example is the logistic equation that was adopted by Rumelhart, Hinton, and Williams (1986), defined below for some output unit *j*:

$$a_j = f(net_j) = \frac{1}{1 + e^{(-(net_j + \theta_j))}} \qquad (7)$$

As is illustrated in Figure 4-1, the logistic function does not break into two parts. Instead, it converts net input into a sigmoid-shaped, continuous function that serves the same role as the Heaviside equation. Low net inputs are converted into near-zero activation, while high net inputs are converted into near-one activation. If net input is exactly equal to θ then the logistic equation returns a value of 0.5. Adopting standard terminology (Ballard, 1986), we will call a perceptron that uses the logistic activation function in its output units an *integration device*.

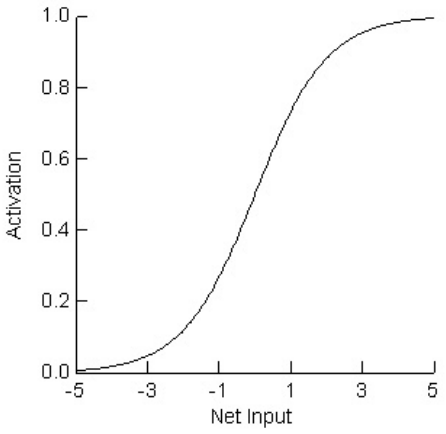

Figure 4-1. The logistic equation produces a sigmoid-shaped activation function.

4.2 Training an Integration Device

4.2.1 The Delta Rule

If one were to insert the logistic activation function into the output unit of a perceptron, then the perceptron will generate continuous responses to stimuli. It will not merely turn on or off. Given its continuous responses in the range between 0 and 1, how would one proceed to train it to perform a task of interest? One approach would be to use the delta rule.

When the delta rule was applied to a perceptron that used the Heaviside activation function, output unit error – the difference between the desired activity and the observed activity – could only adopt one of three different values. If the perceptron turned on when it was supposed to turn off, then error would be equal to -1. If the perceptron made the correct response, then error would be equal to 0. If the perceptron turned off when it was supposed to turn on, then error would be equal to +1.

When the Heaviside equation is replaced with the logistic equation, then the absolute value of error will adopt some continuous value between 1 and 0. That is, it will never be equal to one of three different values as described above. However, because the logistic equation only reaches values of 1 or 0 when net input reaches positive or negative infinity, error will never be exactly equal to1 (or -1) or 0. This is because in our computer simulations of learning, net input will never achieve infinite magnitude.

Because of this, we have to develop an operational definition of successful performance. That is, we have to define some activity value for the output unit that is "high enough" to describe the output unit as being "on". Similarly, we have to define activity that is "low enough" to describe the output unit as being "off".

Once the needed operational definitions are in place, we can again train a network until it has converged to a solution to the presented problem. That is, training comes to a halt when the network generates a "hit" to every pattern.

In the simulations that follow, a "hit" occurs when the integration device turns on – generates activity of 0.9 or higher -- to every pattern that has a target response of 1. A "hit" also occurs when the integration device turns off – generates activity of 0.1 or lower -- to every pattern that has a target response of 0. With this definition of convergence, the delta rule could easily be used to train an integration device.

4.2.2 The Gradient Descent Rule

While the delta rule could be used to train an integration device, this usually does not occur. This is because with a continuous activation function, if one multiplies error (defined as computed for the delta rule) by the first derivative of the activation function, then more efficient learning will result (e.g., Dawson, 2004). The first derivative of the logistic activation function, $f'(net_j)$, is defined as:

$$f'(net_j) = a_j \cdot (1 - a_j) \tag{8}$$

The gradient descent rule for changing a weight in an integration device is defined by multiplying this first derivative into the delta rule equation:

$$\begin{aligned} \Delta w_{ij} &= \eta \cdot a_i \cdot f'(net_j) \cdot \delta_j \\ &= \eta \cdot a_i \cdot a_j \cdot (1 - a_j)(t_j - a_j) \end{aligned} \tag{9}$$

Recall that in this equation, Δw_{ij} is the desired change in the weight from input unit i to output unit j, η is the learning rate, a_i is the activity of the input unit, a_j is the activity of the output unit, and t_j is the desired activity of the output unit.

The gradient descent learning rule can be viewed as a continuous version of the delta rule. That is, it is exactly the same as the delta rule, but has the additional property that it uses information about the continuous activation function to scale the delta rule's error term. When applied to an integration device, the gradient descent rule will lead to very similar results as when the delta rule is used to train a perceptron. This will be demonstrated in the following sections.

4.3 Acquisition Curves for the Integration Device

4.3.1 General Method

The simulations described in the following pages were once again conducted with the Rosenblatt program (Dawson, 2005), available as freeware from: http://www.bcp.psych.ualberta.ca/~mike/Software/Rosenblatt/. The perceptrons trained in all of the examples that follow in Chapter 4 used the logistic activation function in the output unit.

The integration devices were trained with the gradient descent rule with η equal to 0.5. Connection weights were initialized to a random value in the range from -0.1 to 0.1, and θ was initialized to a value of 0. Every training pattern was presented once per sweep in a random order. The integration device's weights, and the bias value of the logistic equation, were updated after every presentation of a training pattern (e.g., Dawson, 2004, p. 181).

4.3.2 Association and Extinction

The first simulation trains an association to CS_A in one phase of learning, and then extinguishes it in a second training phase. Thus, this network consisted of a single input unit and a single output unit. In this experiment, there was only one pattern presented to the integration device during either training phase. The training set (input values along with their target values t_j, as well as trial type in terms of animal learning experiments) is provided in Table 4-1 below:

Phase	Trial Type	Input	t_j
1	A+	1	1
2	A-	1	0

Table 4-1

4.3.2 Results

"The first job of any theory of Pavlovian learning is to account for acquisition curves as a function of the number of training trials" (Miller et al., 1995). When a continuous activation function is combined with a gradient descent learning rule, it would be expected that the network's acquisition curve and extinction curve would have a plausible appearance.

To test this hypothesis, the activation of the output unit was recorded for every Phase 1 learning trial and every Phase 2 extinction trial. The results of a typical network are presented in Figure 4-2 below. For this particular network, 71 learning trials were required to learn to respond to CS_A, and 89 Phase 2 trials were required to extinguish this response. As is evident in the figure, responding to CS_A increases during Phase 1, and the rate of change in responding decelerates. During Phase 2, responding to CS_A decreases, and the rate of change in responding decelerates. These effects are consistent with predictions from the Rescorla-Wagner model. For example, if one sets λ to 1 and α to 0.613 at trial 0, and then sets λ to 0 and α to 0.070 at trial 72, the Rescorla-Wagner model produces a curve that has a correlation of 0.99 with the one in Figure 4-2.

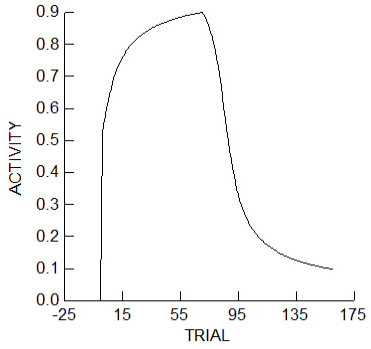

Figure 4-2. Acquisition curves from an example integration device.

Table 4-2 below shows the structure of a typical network at the end of each training phase. Note that the first phase of training resulted in positive (excitatory) network structure, while the second phase of training resulted in negative (inhibitory) structure.

Phase	Unit	Weight	θ
1	CS_A	1.07	-
	Output	-	1.13
2	CS_A	-1.13	-
	Output	-	-1.07

Table 4-2

4.4 On Two Responses

4.4.1 Go or No Go

In many learning experiments, animals are trained to either respond or not to respond. For example, consider the go-no go paradigm used in a recent study of discrimination learning in birds (Lee, Charrier, Bloomfield, Weisman, & Sturdy, 2006).

In this study, when birds were on a "request perch" in front of a feeder they were presented tones of various frequencies. Birds were trained to response to tones from one particular range of frequencies, and to not respond to tones from a different frequency range. In the go-no go paradigm a bird responds by leaving its perch, flying over to a food dispenser to receive a reward. A bird demonstrates that it has learned not to respond to a stimulus by remaining on its perch. Lee et al. (2006) were able to use this paradigm that two different types of chickadees, as well as zebra finches, could learn to discriminate different frequencies of tones.

An integration device that uses the logistic equation as an activation function is well-suited to model experiments in which the network learns either to respond or not. This is because responding can be represented by activity that has a value near one, while not responding can be represented by activity that has a value near zero.

4.4.2 Approach or Avoid

However, there are experiments that involve learning behaviors that are not neatly described as go-no go. For instance, an animal might approach when one stimulus is presented, and to avoid or withdraw when a different stimulus occurs. In such an experiment it is not the case that the animal responds or not. Instead, the animal learns to either respond or to make the opposite response.

For example, Hearst and Franklin (1977) conducted an experiment with pigeons in which an illuminated blank field served as a signal that grain would be delivered, while a black vertical line on a white field signaled that grain would not be delivered. Hearst and Franklin monitored the location of the pigeons. They found at the end of training that when the signal for food was presented, the pigeons moved towards the signal, but when the signal for no food was presented, the birds moved away from the signal.

How would an integration device be used to simulate an experiment in which an animal learns either to approach or to avoid? One possibility would be to use the logistic equation, and to interpret a near zero activity as representing avoidance. The problem with this approach is that it couldn't be used to simulate a situation in which an animal might approach, avoid, or fail to respond at all.

A second approach would be to use two different output units. High activity in one unit could represent approaching, high activity in the other unit could represent avoidance, and low activity in both units could represent failure to respond at all. The problem with this approach is that it doesn't truly reflect the dialectic nature of some responses. In particular, an animal cannot simultaneously move towards a target and away from it. These responses are linked in the sense that they are mutually exclusive. However, this logical link disappears from a perceptron that uses two output units. This is because these output units use completely separate connections. As a result, it is logically possible for the perceptron to turn both of its output units on at the same time, indicating the impossible -- that is, simultaneous approach and avoidance.

A third approach would be to use in integration device that can signal approach, avoidance, and failure to respond with a single output unit. In particular, one could do this by using a sigmoid-shaped activation function the generated values that ranged from -1 to 1. Activity of 1 could represent approach. Activity of -1 could represent avoidance. Activity of 0 could represent a failure to respond. In Section 4.5, an alternative integration device which delivers this kind of responding is introduced.

4.5 The Hyperbolic Integration Device

4.5.1 The Hyperbolic Tangent

Duch and Jankowski (1999) have noted that the logistic activation function that we have used to characterize the integration device can be replaced with a hyperbolic tangent activation function. Like the logistic, the hyperbolic tangent is sigmoid shaped, continuous, and has a simple derivative. The key difference between it and the logistic is that the hyperbolic tangent, illustrated in Figure 4-3, ranges between -1 and +1 instead of between 0 and 1. Therefore this function can be used to define an integration device whose output unit is required to generate negative or positive responses to different stimuli.

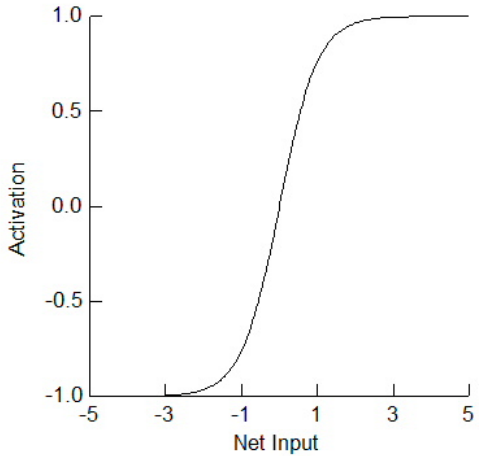

Figure 4-3. The hyperbolic tangent.

Equation 10 defines the hyperbolic tangent as it is used to define the activation function of an integration device:

$$a_j = \tanh(net_j) = \frac{e^{(-(net_j+\theta_j))} - e^{(-(net_j+\theta_j))}}{e^{(-(net_j+\theta_j))} + e^{(-(net_j+\theta_j))}} \quad (10)$$

4.5.2 Hyperbolic Learning Rule

In Section 4.4.2 we saw that a gradient descent learning rule could be used when the activation function was continuous. In a gradient descent technique, more efficient learning is sought by multiplying an output unit's error by the first derivative of its activation function. The first derivative of the hyperbolic tangent activation function, $\tanh'(net_j)$, is defined as:

$$\tanh'(net_j) = \operatorname{sech}^2(net_j) = \frac{1}{\cosh^2(net_j)} = \left(\frac{2}{e^{(-(net_j+\theta_j))} + e^{(-(net_j+\theta_j))}}\right)^2 \quad (11)$$

In this equation, the bias (θ) is the net input value that would produce an activation of 0. That is, while the "threshold" of the logistic equation produces 0.5 activation, the "threshold" of the hyperbolic tangent produces 0.0 activation.

The gradient descent rule for changing a weight in an integration device that uses the hyperbolic tangent is given in Equation 12:

$$\Delta w_{ij} = \eta \cdot a_i \cdot \tanh'(net_j) \cdot \delta_j = \eta \cdot a_i \cdot \operatorname{sech}^2(net_j) \cdot (t_j - a_j) \quad (12)$$

In this equation, Δw_{ij} is the desired change in the weight from input unit *i* to output unit *j*, η is the learning rate, a_i is the activity of the input unit, a_j is the activity of the output unit, and t_j is the desired activity of the output unit. Note that this equation is identical to Equation 9, with the exception that a different derivative is used, and a_j is computed by using the hyperbolic tangent equation that was provided in Equation 10.

In the next section, we will see how this new learning rule for this variant of the integration device can be used to train a network to approach or to avoid, depending upon the signals that are provided.

4.6 Approach or Avoid

4.6.1 Inhibition and Excitation

We saw earlier that Hearst and Franklin (1977) conducted an experiment in which animals learned to approach when presented one stimulus, and to retreat when presented another. The simulation below demonstrates this type of associative learning in an integration device that employs the hyperbolic tangent, and demonstrates its ability to produce three different responses to three different stimuli. The point of training is to generate a positive response when CS_A is presented, a neutral or zero response when CS_B is presented, and a negative response when CS_C is presented.

4.6.2 Training Set

Three patterns are presented to a perceptron during a sweep of training. They, along with their target values t_j, are provided in Table 4-3 below. The key difference between this table and those that we have seen previously is that the value of t_j for Pattern 3 is -1. An integration device that employs the logistic function is not capable of generating this response, and so would be unable to learn the training set given in Table 4-3. However, this response is completely compatible with the hyperbolic tangent.

Pattern	CS_A	CS_B	CS_c	t_j
1	1	0	0	1
2	0	1	0	0
3	0	0	1	-1

Table 4-3

4.6.3 Results

25 different integration devices, each started randomly according to the procedure described in Section 4.3.1, served as "subjects". A key change in methodology in this simulation was that the perceptrons used the hyperbolic tangent (Equation 10) as the activation function, and were trained with the gradient descent rule for this function (Equation 12). Each integration device learned to respond correctly to each pattern in the training set (that is, to generate the desired outputs (t_j) provided in Table 4-3). On average, this required 27.3 sweeps of training. The structure of a typical network after training completed is provided in Table 4-4.

Unit	Weight	θ
CS_A	1.44	-
CS_B	-0.04	-
CS_C	-1.53	-
Output	-	0.05

Table 4-4

Table 4-4 reveals how each CS on its own activates the output unit after training completed. When CS_A has a value of 1, it sends a signal of 1.44, producing a net input of 1.49 when combined with θ of 0.05. This is a net input that is positive enough to turn the output unit on. Similarly, when CS_B is activated, it in combination with θ produces a net input of 0.01. This is close enough to 0 to cause the output unit to generate zero activity. Finally, when CS_C is activated, it in combination with θ produces a net input of -1.48. This is negative enough to cause the output unit to generate activation of -1.

There are two additional types of observations to make from Table 4-4. First, the three connection weights reflect the functional role of each stimulus. That is the excitatory stimulus (CS_A) is associated with a positive (excitatory) weight. The neutral stimulus (CS_B) is associated with a near zero (neutral) weight. The inhibitory stimulus (CS_A) is associated with a negative (inhibitory) weight.

Second, the bias of the hyperbolic tangent is near zero. It will be argued later in this manuscript (Section 6.2) the bias of an activation function can be considered to represent the associative strength of the background context of an experiment. Experimental results have indicated that the association involving the context should not be inhibitory (e.g., Bouton & King, 1983). The network above illustrates the ability to produce an inhibitory response in which the bias is not inhibitory (negative). Later this will become an issue because in some simulations that use the logistic activation function the performance of the network requires an inhibitory bias.

4.7 Conditioned Inhibition in the Integration Device

4.7.1 Conditioned Inhibition

One important phenomenon studied by Pavlov (1927) was conditioned inhibition. In conditioned inhibition, an agent learns that some CS_A signals either the absence of reward or the presence of some aversive conditions.

One approach to measuring conditioned inhibition is testing "retardation of acquisition" test (e.g., Hammond, 1968; Rescorla, 1969). This involves two groups of subjects, experimental and control. The control group learns an excitatory response to CS_A. The experimental group is involved in two phases of training. First, it learns an inhibitory response to CS_A. Then it undergoes the same training that the control group received. If CS_A has become a conditioned inhibitor for the experimental group, then it should take longer for this group to learn an excitatory response to this stimulus than was the case for the control group.

4.7.2 Method

Can conditioned inhibition be observed in an integration device? To answer this question, the retardation of acquisition paradigm was adopted in a simulation experiment. Table 4-5 provides the training sets used in the two phases for the experimental group; the control group only was trained on the Phase 2 pattern.

Phase	Pattern	CS_A	t_j
1	1	1	-1
2	1	1	1

Table 4-5

In this simulation, 25 different integration devices served as "subjects" in each condition. All of these networks used the hyperbolic tangent as an activation function, and were initialized and trained according to the general procedures described earlier. In the experimental condition, integration devices were first trained to convergence on the Phase 1 patterns. Then, they were trained – without reinitializing weights – to convergence on the Phase 2 patterns. The control group networks were simply trained to generate hits to the Phase 2 training patterns. The dependent measure of interest was the number of sweeps required to converge to the Phase 2 patterns for both groups of integration devices.

4.7.3 Results

If conditioned inhibition was produced during the first phase of learning, then learning the Phase 2 response should take significantly more epochs for the experimental condition than for the control condition. This was indeed the main result of this study. On average, control perceptrons generated a "hit" in Phase 2 after 32.6 sweeps of training, while experimental condition perceptrons required an average of 41.3 sweeps to achieve this level of performance. This result was statistically significant ($t = 48.52$, $df = 48$, $p < 0.0001$).

How is this result produced? When training of a typical control network is finished, the weight between the output unit and the input unit that represents the presence of CS_A is 1.47, and the output unit's bias (θ) is 0.01. On average, such a configuration is achieved after 32.6 sweeps, when the connection weight starts in the range between 0.10 and -0.10. In contrast, after Phase 1 training a typical experimental network has a connection weight of -1.46 and θ of -0.02. Because this connection weight is so negative, additional training is required to convert it to a positive value that is required to end Phase 2 training.

4.8 Summation in the Integration Device

4.8.1 The Summation Test

An alternative method to measuring conditioned inhibition is summation (e.g., Rescorla, 1969, 1971; Solomon et al., 1974).). In the summation technique, a stimulus was first trained to become a conditioned inhibitor. Then it was paired with a second stimulus that was known to be an excitor. If the response to the pair of stimuli was less than the response when the excitor was presented alone then this indicated that the training had succeeded in converting the initial stimulus into a conditioned inhibitor.

The simulation below provides a variation of this approach. First, a network was trained on the same training set that was used in Section 4.6, and was presented in Table 4-3. That is, an integration device that used the hyperbolic tangent learned to generate a response of 1 to CS_A (excitor), a response of 0 to CS_B (neutral stimulus), and a response of -1 to CS_C (inhibitor). Because each of these stimuli was associated with its own input unit and its own connection weight in the network, and because each stimulus was always presented in isolation, these three different types of associations (excitatory, neutral, and inhibitory) were represented and developed independently.

After this training, the network's response to two combined stimuli (CS_A + CS_B, and CS_A + CS_C) was measured without any additional training occurring. If CS_C had become a conditioned inhibitor, then through summation the network's response when it was paired with CS_A should be much less than when CS_A was paired with the neutral stimulus CS_B.

4.8.2 Results

25 integration devices served as "subjects" in this study. As was the case in Section 4.6, they all converged to correct responses to the training set; on average this occurred after 27.3 sweeps of training. After this training a typical network generated a response of 0.90 to CS_A, a response of 0.0 to CS_B, and a response of -0.90 to CS_C.

The summation test for conditioned inhibition was conducted by presenting pairs of stimuli to the networks after training was complete. When CS_A was presented in conjunction with the neutral stimulus CS_B, the average network response was 0.90. In contrast, when CS_A was presented in conjunction with CS_C, the average network response declined dramatically 0.00. This difference in responding was statistically significant (t = 199.76, df = 48, $p < 0.0001$).

4.9 Simulating Renewal

4.9.1 Renewal

One phenomenon of considerable interest is renewal (Bouton & Bolles, 1979). In the renewal paradigm, conditioning to a stimulus is conducted within one context. Then, this conditioning is extinguished – but in a different context. After this extinction, the effectiveness of the stimulus is tested in the original context. When renewal is observed, the extinguished stimulus recovers some of its effectiveness in the original context.

If one operationalizes context using the presence or absence of additional stimuli, then it is possible to study renewal in the integration device: In Phase 1 of training, an integration device is trained to respond when CS_A and CS_B are both presented (i.e., AB+ conditioning). In Phase 1 CS_A is the stimulus that is targeted by conditioning, and CS_B provides a context. In Phase 2, conditioning to CS_A is extinguished, but in a different context: extinction occurs when a new stimulus, CS_C, is present (i.e., AC- conditioning). The response of the integration device to A in the original context (CS_A and CS_B) is then examined. If renewal is observed, then the response to this stimulation will be greater than the response in the extinction context (CS_A and CS_C), even though CS_A has been involved in extinction trials.

Renewal was tested by employing the paradigm described in the preceding paragraph, using the training patterns provided in Table 4-6. During Phase 1, only one pattern was presented, and the integration device was trained to convergence. During Phase 2, the connection weights produced by Phase 1 training were maintained, and the integration device was trained to convergence (in this case, extinction) again. At the end of Phase 2, the responses of the integration device were recorded to three different stimuli (AB, AC, and A). This permitted the response of the network in the original context, in the extinction context, and in a novel context (in which neither CS_B nor CS_C were present) to be recorded

Phase	Type	CS_A	CS_B	CS_c	t_i
1	AB+	1	1	0	1
2	AC-	1	0	1	0
Posttest	AC	1	0	1	-
	A	1	0	0	-
	AB	1	1	0	-

Table 4-6

4.9.2 Results

The results demonstrated renewal in the integration device. When CS_A was presented in the extinction context (AC), the average response of a perceptron was 0.10. However, when CS_A was presented in the original context (AB), the average response was much higher (0.44). When CS_A was presented in the novel context (A), the average response was between that observed in the other two conditions (0.27). A one-way ANOVA indicated that the effect context on recall was statistically significant ($F = 9085.6$, $df = 2, 72$, $p < 0.0001$). Post hoc comparisons between the three means revealed that each was significantly different from the other at $p < 0.0001$.

This demonstration of renewal is a special case of conditioned inhibition. In the extinction phase, part of the reduction in the network's response was due to a decrease in the association to CS_A, but part was due to CS_C becoming a conditioned inhibitor. When CS_A was presented in the original context, this conditioned inhibitor was not present, and the integration device generated a stronger-than-expected response to CS_A.

That renewal in the network depends on a context becoming inhibitory could be argued by some to be a failure, because experimental results are incompatible with this possibility (e.g., Bouton & King, 1983). However, it has recently been argued that inhibitory associations involving contexts are crucial to explaining many results, and are not easily detected experimentally because of attentional factors (Schmajuk, Larrauri, & Labar, 2007). The issue of inhibition and context is revisited later in the manuscript.

4.10 Superconditioning in an Integration Device

4.10.1 Superconditioning

Previously, we have seen that perceptrons can demonstrate blocking and conditioned inhibition. Therefore we should expect that another phenomenon can be produced in the perceptron: superconditioning. This is because superconditioning can be viewed as the "mirror image" of blocking, and is dependent upon conditioned inhibition (Williams & McDevitt, 2002).

Recall from Section 3.8 that blocking occurs when a previously learned excitatory association involving a CS disrupts the ability to create a new association involving a different CS. In comparison, superconditioning occurs when a previous learned conditioned inhibitor enhances the creation of a new association involving a new CS.

To produce superconditioning, two phases of classical conditioning are used. The first phase involves CS_A and CS_B, and is designed to make CS_B a conditioned inhibitor. The second phase involves reinforcing a third stimulus, CS_C. In a control condition, CS_C is paired with a fourth stimulus, CS_D, which has not been involved in any previous training. In an experimental condition, CS_C is paired with the conditioned inhibitor CS_B. If superconditioning occurs, then the association produced for CS_C in the experimental condition should be enhanced relative to the association produced for the same stimulus in the control condition.

4.10.2 Method

Table 4-7 provides a training set involving four CSs that was used to study superconditioning in an integration device. The learning rate and the starting values for connection weights and the output unit's bias were the same as those used in the previous simulations in this chapter. A network was first trained in Phase 1 to produce conditioned inhibition to CS_B. Then the network was trained in Phase 2 using either the control or the experimental training patterns. During this second phase, the connection weights were examined after every 5 sweeps of training until the network converged. If superconditioning occurred, then it should be evident in an examination of the changing connection weights.

Phase	CS_A	CS_B	CS_c	CS_D	t_j
Phase 1	1	0	0	0	1
	1	1	0	0	0
Phase 2 Control	0	0	1	1	1
Phase 2 Experimental	0	1	1	0	1

Table 4-7

4.10.3 Results

This study indicated that superconditioning can be produced in an integration device. In the control condition, CS_C and CS_D both typically developed moderately positive weights at a moderate rate. In the experimental condition, the connection weight for CS_C usually grew to more than twice than what was observed in a control network, at a more extreme rate.

This is illustrated in Figure 4-4. The solid bars illustrate the changing weight of the connection between CS_C and the output unit for one network during Phase 2 in the control condition. The other bars illustrate the same connection weight for a different network during Phase 2 training in the experimental condition. Clearly when CS_C is paired with the conditioned inhibitor, it develops a much stronger positive connection weight, and does so far more rapidly, than when CS_C is paired with CS_D. This result was easily replicated in a number of different networks.

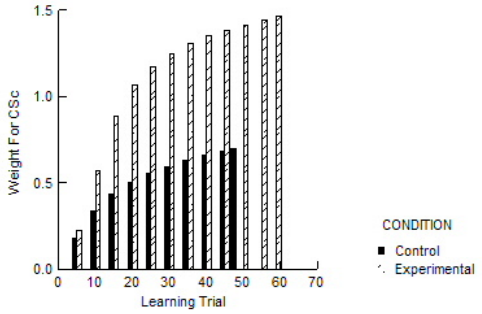

Figure 4-4. Weight changes for CS_C over time for a control and for an experimental network.

4.11 Positive Patterning in the Integration Device

4.11.1 Positive Patterning

In Section 3.6 we described a traditional perceptron that was trained to respond to a compound stimulus, but was also trained not to respond to the components of this stimulus when the components were presented separately. The purpose of Section 4.7 is to examine the responses of an integration device when taught in this positive patterning paradigm.

In this simulation, three different patterns were presented to the perceptron to simulate a typical positive patterning experiment. These stimuli defined the learning conditions A-, B-, and AB+. The relationships between these trial types, the input patterns, and the associated responses of the perceptron, are provided below in Table 4-8.

Trial Type	CS_A	CS_B	t_j
A-	1	0	0
B-	0	1	0
AB+	1	1	1

Table 4-8

4.11.2 Results

25 different integration device "subjects" were trained to convergence on the three different patterns provided in Table 4-8. On average, a network converged after 964 sweeps. An example network, which converged in 967 sweeps, generated 0.08 to the first two patterns, and 0.9 to the third at the end of training. The structure of this network is provided in Table 4-9.

Unit	Weight	θ
CS_A	4.59	-
CS_B	4.59	-
Output	-	-6.98

Table 4-9

This pattern of connectivity solves the positive patterning problem as follows: When only one of the CSs is present, the net input that is passed onto the logistic activation function is negative. This is because an input unit signal of 4.59 is combined with the strongly negative bias of -6.98 to produce a net input of -2.39. This net input is low enough for the logistic function to return a near zero activity. However, if both of the CSs are present, the two positive input unit signals compensate for the negative bias and produce an overall net input of 2.2. This value is high enough to be converted into high activity by the logistic equation.

It is important to note that the integration device converged to a positive patterning solution much slower than did the perceptron described in Section 3.6. The reason for this is that for a continuous activation function, the output unit's error is also continuous. In contrast, the error for a perceptron that uses a Heaviside activation function is digital: it will be equal to -1, +1, or 0. With a much more restricted possibility for error, a device's search for a configuration that will reduce error is highly constrained, and learning is more efficient.

The difference in learning speeds for the two types of perceptrons, however, is not particularly meaningful. Many different manipulations can be used to change learning speeds for one type of network that is learning the same type of problem. For instance, the number of sweeps to converge to a solution is highly dependent upon the learning rate. At best, the number of sweeps required for a network to converge is a casual measure of problem difficulty.

4.12 Associating To Compounds and Their Components

4.12.1 Stimulus Generalization

One central phenomenon in the animal learning literature is stimulus generalization (Rescorla, 1976). Stimulus generalization occurs when one stimulus generates a response because it shares some of the properties of another stimulus, where this second stimulus is capable of causing a response to occur.

The Rescorla-Wagner model is one extremely influential theory of animal learning, and it is described in detail in the next chapter. For now, suffice it to say that it was not explicitly designed to account for stimulus generalization effects. However, it does permit one to operationalize stimulus similarity in terms of the number of shared components. That is, the stimulus AC is similar to the stimulus AB in the sense that both share a single component (A).

When this perspective on stimulus similarity is adopted, the Rescorla-Wagner model can make some interesting and counterintuitive predictions (Rescorla, 1976). Consider an animal that is trained to respond to a compound stimulus (i.e., AB+). According to the Rescorla-Wagner model, this training would increase the associative strengths to both CS_A and CS_B. However, when training on the compound proceeds long enough, the two associative strengths would achieve maximum levels, and further training on the compound would not affect them.

However, once the learner has achieved maximum learning to the compound, the Rescorla-Wagner model predicts that the strengths of the individual associations can still be increased by a different type of training. In particular, if one continues to train using only one of the components of the compound stimulus, then the associative strength for that component will increase. That is, if one trains AB+ to asymptote, and then continues to train AB+, then the association for CS_A will not grow. However, if one trains AB+ to asymptote, and then continues to train on a different stimulus that contains A as a component, then the association for CS_A *will* grow.

The purpose of the current simulation study was to determine whether this counterintuitive effect could be produced in an integration device. All perceptrons were trained to convergence on a single pattern, AB+ (see Table 4-10). In the control condition, training on AB+ continued. In the experimental condition, when training continued a different pattern (AC+, see Table 4-10) was used. 25 perceptrons served as "subjects" in each condition. If the integration device responds in accordance with the Rescorla-Wagner model, then the weight of CS_A should increase in the experimental condition, but not in the control condition

Trial Type	CS_A	CS_B	CS_C	t_j
AB+	1	1	0	1
AC+	1	0	1	1

Table 4-10

4.12.2 Results

Figure 4-5 presents the average weight of CS_A in the different phases of the experiment. These results are consistent with the Rescorla-Wagner model's prediction: this weight does not change significantly in the control condition, but does grow – on average by 32.4% -- when the network receives continued training on a new stimulus that contains A as one of its components.

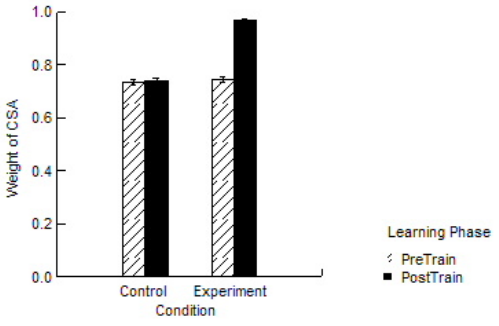

Figure 4-5. Stimulus generalization in an integration device.

4.13 Overshadowing in the Integration Device

4.13.1 All Components Are Not Equal

Pavlov (1927) reported the results of experiments with compound stimuli in which the intensities of the stimulus components were manipulated. For example, in one study an animal was conditioned to respond to a pair of tones that were of equal intensity. After conditioning, the individual tones were presented, and were found to elicit the same conditioned response as did the compound stimulus. Different results were obtained when the compound stimulus was comprised of tones of unequal intensities. The higher intensity tone elicited the same response as the compound CS. However, "the effect of the tone of weaker intensity when tested singly was now very small or absent altogether" (p. 145).

This latter phenomenon is known as overshadowing. Overshadowing occurs when a stronger stimulus overshadows or even prevents learning about a weaker stimulus that is present at the same time. It is an example of how the ability of a stimulus to elicit behavior does not depend on the stimulus itself, but also upon the context in which the stimulus is presented.

One can attempt to explore overshadowing in an integration device by manipulating the intensity of the input unit representation of CSs. Table 4-11 presents a training set in which CS_A is ten times less intense than CS_B when both are presented simultaneously to serve as a compound stimulus:

Type	CS_A	CS_B	t_j
AB+	0.1	1	1

Table 4-11

4.13.2 Results

The training set in Table 4-11 was used to train 25 different perceptrons as subjects. On average, convergence was achieved after 71 epochs. After this training, the average response to the individual presentation of CS_B was 0.90, while the average response to the individual presentation of CS_A was 0.75. While this difference was small, it was statistically significant ($t = 133.75$, $df = 48$, $p < 0.0001$). This difference in responding indicates that the more intense CS_B overshadowed the less intense CS_A.

Stronger evidence for overshadowing comes from an examination of the connection weights of an integration device trained on this compound stimulus. Table 4-12 presents the connection weights of a typical network after it has converged. Note that the weight associated with CS_B is an order of magnitude larger than the weight associated with CS_A.

Unit	Weight	θ
CS_A	0.19	-
CS_B	1.07	-
Output	-	1.11

Table 4-12

Compare this situation to one in which a network is trained on CS_A alone (with an intensity of 0.1; the activity of CS_B is 0). With this training pattern, the response of the integration device to CS_A alone will be 0.9 (this produces a "hit" to stop training). The weights of a typical network are presented in Table 4-13. In this case, the weight associated with CS_A is an order of magnitude greater than the weight associated with CS_B.

Unit	Weight	θ
CS_A	0.13	-
CS_B	-0.05	-
Output	-	2.19

Table 4-13

One question to answer is why the overshadowing effect seems so much larger when connection weights are considered than when output unit responses are considered. Later in this book we will argue that simulated experiments of the type that we have been considering require an additional stimulus, called a null stimulus. We will see that when the null stimulus is added to the overshadowing paradigm, much more dramatic evidence of overshadowing is achieved.

4.14 Recovery from Overshadowing

4.14.1 The Comparator Hypothesis

An important alternative to the Rescorla-Wagner model is Miller and Matzel's (1988) comparator hypothesis. which can be described as a variant of contingency theory (e.g. Rescorla, 1968, 1969; see also Cheng & Holyoak, 1995). According to comparator theory, two sources of information compete for control of a response. One is the CS. The other is a "comparator stimulus" -- background stimuli or discrete cues that were present when the CS was trained. Only if the CS is a better predictor of the US than the comparator stimulus is the CR is exhibited.

In general, empirical support for the comparator hypothesis is obtained by performing post-training manipulations to the associative strength of the comparator stimulus. If this strength is decreased, then the model predicts that excitatory responding to the CS should consequently increase.

Consider an overshadowing experiment in which AB+ training occurs, but CS_A is weak (e.g., activity = 1) compared to CS_B (e.g., activity = 10). The comparator hypothesis predicts that if CS_B is extinguished, then CS_A should recover from overshadowing, because the extinction of CS_B reduces its ability to compete against CS_A. Several experiments have demonstrated such recovery from overshadowing (Kaufman & Bolles, 1981; Matzel, Schachtman & Miller, 1985; Matzel, Shuster & Miller, 1987).

The Rescorla-Wagner model cannot explain recovery from overshadowing when the overshadowing CS_B is extinguished. This is because the model explains overshadowing by appealing to the fact that CS_B prevents the overshadowed CS_A from developing an association that is sufficiently excitatory. Extinguishing CS_B will have no effect on the associative strength of CS_A.

4.14.2 A Model Variation

The integration devices that we have been describing will also fail to generate recovery from overshadowing, for the same reasons that face the Rescorla-Wagner model. However, a minor change to input coding avoids this limitation. The learning rules that we have been using require a non-zero input unit activity in order for a connection weight to change. When an absent CS is coded with a 0, its association will not be altered. However, if an absent CS was coded with a -1 instead of a 0 (see also Dickinson & Burke, 1996), then the weight of CS_A can change when CS_B is extinguished, and recovery from overshadowing is possible. This coding change is equivalent to saying that the nonpresentation of a CS is as informative as its presentation, a fact which has been established in some experimental studies (e.g., Van Hamme & Wasserman, 1994).

Consider the training set in Table 4-14. In Phase 1, AB+ training produces overshadowing. In Phase 2, B- training extinguishes the overshadowing CS.

Phase	Type	CS_A	CS_B	t_j
1	AB+	1	10	1
2	B-	-1	10	0

Table 4-14

For a typical network, Phase 1 learning is completed after 6 sweeps, and overshadowing has occurred. One example network generated a response of 0.43 to CS_A and 0.92 to CS_B after this training. Phase 2 training extinguished CS_B; the network now generated a response of 0.10 to this stimulus. However, this extinction resulted in a mild recovery in the response to CS_A , which produced output activity of 0.54 to this stimulus now (instead of 0.23).

The Rescorla-Wagner model can be modified to produce this kind of effect (see also Van Hamme & Wasserman, 1994). For example, in Section 5.4.2 a unified Rescorla-Wagner equation is presented. The assumption in this equation is that an absent CS is encoded with a 0 (see Equation 23 below in Section 5.4). However, if this assumption is changed, and an absent CS is represented by a non-zero value such as -1, then this equation should be capable of generating recovery from overshadowing due to the extinction of the overshadowing stimulus.

4.15 Blocking and the Integration Device

4.15.1 The Blocking Paradigm

Overshadowing is an illustration that different CSs compete with one another to gain access to the mechanisms that modify associative strength. Another example of this is a phenomenon called blocking (Kamin, 1968, 1969).

Blocking is demonstrated as follows: For an experimental group of animals, CS_A is conditioned to elicit a response. Then, CS_A and CS_B are presented as a compound stimulus to the animals that have already learned the association from CS_A. A control group of animals is only trained using the compound stimulus. At the end of training, both groups of animals are tested by measuring the response to CS_B. With this paradigm, the experimental group shows less of a response to CS_B than does the control group. It is as if the pre-training with CS_A blocked the later learning that could have occurred with CS_B.

An integration device can be trained in the blocking paradigm by creating two different training sets, each consisting of a single pattern. In the first phase of the experimental condition, only one input unit (CS_A) is trained to produce a response in the output unit. After this network converges, its connection weights are preserved, and it is trained on a compound stimulus (CS_A and CS_B) until convergence is achieved again. In the control condition, a network is only trained to converge on the Phase 2 pattern. The two training patterns are presented in Table 4-15.

Phase	Type	CS_A	CS_B	t_i
1	A+	1	0	1
2	AB+	1	1	1

Table 4-15

4.15.2 Results

The standard test of blocking is to compare the network's response to CS_B in the control condition to the response to the same stimulus in the experimental condition. When this is done, a small blocking effect is observed. On average, the control networks generate a response of 0.81 to CS_B. In contrast, experimental networks generate a response of 0.76 to CS_B. Though this difference in mean responding is small, it is statistically significant (*$t = 24.72$, $df = 48$, $p < 0.0001$*).

A more dramatic measure of blocking is achieved when the structure of the networks are examined. Table 4-16 presents the final structure of a typical control and a typical experimental network. Note the difference in the weights for CS_B in the two networks. In general, the blocking paradigm produces extremely small connection weights for the blocked stimulus in comparison to control networks.

Condition	Unit	Weight	θ
Control	CS_A	0.68	-
	CS_B	0.77	-
	Output	-	0.75
Experimental	CS_A	1.13	-
	CS_B	-0.03	-
	Output	-	1.11

Table 4-16

Again, one question to answer is why the blocking effect seems so much larger when connection weights are considered than when output unit responses are considered. We will see that when a null stimulus is added to the blocking paradigm, much more dramatic evidence of blocking is achieved. In the current study, the absence of the null stimulus produces a weaker effect. Nonetheless, both the responses and the connections of the trained integration devices are consistent with the claim that this type of artificial neural network can demonstrate blocking.

4.16 Success from Failure

4.16.1 Rescorla-Wagner Limitations

Rescorla and Wagner's (1972) learning model has been enormously influential and successful. However, there are a large number of phenomena that it fails to capture (e.g., Miller, Barnet & Grahame, 1995). The simulation results that have been presented to this point have illustrated that simple artificial neural networks can generate many of the successes captured by the Rescorla-Wagner formalism. If these two approaches are strongly related, then the networks should exhibit these failures.

4.16.2 Network Limitations

Indeed, there are a number of associative learning phenomena that the networks that have been described cannot produce. These failures are shared by the Rescorla-Wagner model, and therefore count as successes in the task of exploring commonalities between the two approaches.

For example, network weights are only changed when the learning rule is applied. It is impossible to alter the weights without training the network. As a result, the perceptrons cannot account for such effects as spontaneous recovery from extinction (e.g., Pavlov, 1927), where the behavior of an agent changes in the absence of additional training. For similar reasons the networks cannot account for spontaneous recovery from overshadowing (e.g., Kraemer, Larivier, & Spear, 1988).

Other failures result because network training has specific and invariable effects on connection weights. For example, the network produces overshadowing (Section 4.14) because the salient stimulus disrupts the establishment of the weaker stimulus' connection weight. The Rescorla-Wagner model has a similar account of overshadowing. As a result, neither type of model can explain why in some cases large numbers of trials in which both the weak and strong stimuli are reinforced as a compound result in the elimination of overshadowing (e.g., Bellingham & Gillette, 1981). As well, neither approach can explain potentiation (e.g., Clarke, Westbrook & Irwin, 1979), which is a phenomenon in there is enhanced responding to a weak CS that is paired with a more salient CS. Both the network model and the Rescorla-Wagner model predict overshadowing instead of potentiation in this type of paradigm.

Other shared failures can be demonstrated with simulations. For example, Hall, Mackintosh, Goodall and Dal Martello (1977) trained a weak CS to be a strong predictor of a US. Later, additional training was performed where the weak CS was also paired with a stronger CS. As training proceeded, the weak CS lost control of behavior. This is not predicted by the Rescorla-Wagner, which instead predicts that the established association involving the weak CS should overshadow the more salient CS during later training. An integration device makes the same (failed) prediction. In one simulation, a weak CS_A (0.1 activation) was consistently reinforced, and at the end of training had adopted a very strong positive connection weight (43.97). Subsequently CS_A was paired (and reinforced) with a much more salient CS_B (1.0 activation). The prior learning involving CS_A blocked learning involving CS_B which only developed a connection weight of 0.02.

This is far from an exhaustive list of results for which the networks generate results that are not in agreement with those obtained from animal experiments. Other examples will appear in later chapters. That perceptrons have such limitations should be of no surprise, because limits to their computational power (e.g., Minsky & Papert, 1969) have been used to argue for their limits in simulating associative learning (e.g., Pearce, 1997). There are two points of importance to note concerning network failures. First, the failures listed in this section are shared with the Rescorla-Wagner model, providing further evidence of the relationship between the two that will be detailed in the next chapter. Second, in Chapter 7 we will see some divergences in failures – Rescorla-Wagner failures turned to network successes, and vice versa – that will raise important questions about how the two should be equated.

4.17 From the Empirical to the Formal

4.17.1 The Perceptron in Review

In the past three chapters, we have introduced the properties of some simple artificial neural networks. One way to describe these systems is as a functional model of the information processing carried out by a neuron that is receiving signals from others.

A second description of these networks is as an account of how associations involving conditioned stimuli change during classical conditioning. From this second perspective, the input units represent the presence or absence of conditioned stimuli, and the connection weights in the network represent the strengths of the associations that are modified as a result of conditioning.

If this second account is plausible, then one should be able to use a learning rule to train perceptrons to respond appropriately to patterns that represent the conditions of paradigms from the animal learning literature. Chapters 3 and 4 have presented a number of empirical results that have supported this hypothesis. In particular, the following aspects of conditioning have been demonstrated in the perceptron:

- Classical conditioning of individual stimuli
- Plausible acquisition and extinction curves
- The effect of CS intensity on the rate of conditioning
- The effect of US intensity on the rate of conditioning
- Associations to compound stimuli
- The discrimination of compound stimuli from their components
- Overshadowing
- Blocking
- Conditioned inhibition (both in terms of retardation of acquisition and in terms of summation)
- Renewal, or context-dependent extinction
- Superconditioning

In addition, it was noted in Section 4.16 that the networks share a number of limitations with the Rescorla-Wagner model. That is, there are a number of conditioning phenomena that are not captured by either type of model.

4.17.2 Comparing Learning Theories

These results suggest that there is a great deal of commonality between the view of learning that comes from the study of classical conditioning, and the view of learning that comes from the study of perceptrons. In the next two chapters we will be exploring the relationship between these two approaches.

Fortunately, in addition to empirical studies of learning in machines and animals, there has been a great deal of formal analysis in these two domains. This provides us with an opportunity to perform a computational analysis of the two types of learning. In particular, we can examine the equations that govern perceptron learning, and the equations that have been proposed to account for animal learning, and determine the extent to which they are related.

This is the goal of Chapter 5. It will briefly review the mathematics of a particularly influential theory of learning, the Rescorla-Wagner model (Rescorla & Wagner, 1972). It will then present proofs that show that this model can be translated into the mathematics of the delta rule that has been used to train the networks described in Chapters 3 and 4.

Interestingly, the formal equivalence that will be proven in Chapter 5 will lead us to a paradox that will be introduced in Chapter 7. The empirical results of Chapters 3 and 4, and the proofs to be given in Chapter 5, would lead us to believe that any phenomenon that is predicted by the Rescorla-Wagner model should also be evident in the training of a perceptron. Chapter 7 will present several interesting results that show that this expectation is false! We will then have to resolve this paradox by revisiting the relationships between the two views of learning.

Chapter 5: From Animal Learning to the Delta Rule

The previous chapters have related network learning to animal learning by examining the performance of perceptrons (old and modern) in a number of associative learning experiments. This chapter explores a more formal relationship between these two approaches to learning formally. The chapter begins by introducing an influential model of animal learning, the Rescorla-Wagner model. It then reviews older results that proved that the delta rule and the Rescorla-Wagner equation are identical if one assumes that a perceptron's activity is the same as its net input. Finally, the chapter elaborates these earlier results to show the relationship between the two learning rules holds when the perceptron employs a nonlinear activation function.

A peer-reviewed monograph published by Comparative Cognition and Behavior Reviews on behalf of the Comparative Cognition Society.

5.1 Association from Expectation

Phenomena like overshadowing and blocking indicated that simple relationships between concepts, such as mere contiguity, could not account for the strength of the association between them. Associations appear to be determined largely by the predictive information provided by stimuli (e.g., Egger & Miller, 1963). For example, in the blocking paradigm CS_B does not provide any prediction of the occurrence of the US beyond the prediction that is already provided by CS_A. As a result, no associations involving CS_B are established.

One general account of such learning explicitly assumes that the CS is a source of information that provides expectations about the occurrence of the US (Rescorla & Wagner, 1972). If the expectations are correct, then no learning occurs. However, if there is surprise – if the expectations are not correct – then associations are modified in such a way to improve the predictive power of the CS when it is next encountered.

Rescorla and Wagner (1972) provided an extremely influential formalization of this notion. To begin, let us assume that we are dealing with a single CS and a single US. The strength of the association between these two stimuli at any time *t* is symbolized as $V_{(t)}$. Rescorla and Wagner were concerned with specifying how this associative strength changed over time, which can be represented as $\Delta V_{(t)}$.

Rescorla and Wagner (1972) assumed that the repeated pairing of a US and a CS would result in a gradual increase in the association between them, up to a maximal value (λ) that was determined by the magnitude of the US. They also included a parameter (α) that represented the salience of the CS, which is also known to affect conditioning (Pearce, 1997). A second parameter (β) was also included as a learning rate, and was associated with the US, under the assumption that different USs would lead to different rates of learning. Both α and β were assumed to be constants that could range in value from 0 to 1.

Rescorla and Wagner (1972) proposed that the amount of change in an association at any given time was a function of the difference between the current associative strength ($V_{(t)}$) and the maximum possible associative strength (λ). Informally, this difference reflected the predictive power of the CS. If this difference was large, then the predictive power of the CS was poor, and the strength of the association required modification. If this difference was small, then the predictive power of the CS was excellent, and little modification of the association was required.

Formally, Rescorla and Wagner (1972) defined the change in associative strength that was required at any time *t* as:

$$\Delta V_{(t)} = \alpha\beta\left(\lambda - V_{(t)}\right) \tag{13}$$

With the change in associative strength defined, Rescorla and Wagner could write an iterative equation that defined how the associative strength between the US and the CS changed over a sequence of pairings:

$$V_{(t+1)} = V_{(t)} + \Delta V_{(t)} \tag{14}$$

This simple equation leads to some interesting predictions. For example, the amount of change in an association is a function of the difference between λ and $V_{(t)}$. At the start of learning (when there is assumed to be no association between the US and the CS), this value will be large, and will result in large changes in associative strength. However, every time the association is modified, the result will be to decrease the difference between λ and $V_{(t)}$. As a result, this equation for learning predicts that the amount of learning that occurs will slow down as $V_{(t)}$ grows. Indeed, if one plots $V_{(t)}$ as a function of *t*, the result is a decreasing exponential function that levels off to the maximum value of λ. This is in nice agreement with observations of animal learning (e.g., Pearce, 1997, p. 28).

5.2 Associations from Multiple Predictors

5.2.1 The Rescorla-Wagner Model

One of the motivations for the Rescorla-Wagner model was to develop a mathematical model of learning that was capable of handling such phenomena as blocking. This requires that the model from Section 5.1 be extended to handle a situation in which more than one CS is present.

To do this, Rescorla and Wagner (1972) proposed that the overall associative strength that was being compared to λ was actually the sum of individual associative strengths. Each of these individual strengths was the association between the US and a different CS.

For example, consider a situation in which three different conditioned stimuli, CS_A, CS_B, and CS_C, were present. The association between each of these stimuli and the US can be represented as V_A, V_B, and V_C respectively. The overall associative strength (ΣV) for this example can then be defined as:

$$\Sigma V = V_A + V_B + V_C \tag{15}$$

After defining ΣV, Rescorla and Wagner (1972) were in a position to use it to define the change in individual associative strengths that would occur at some time *t* during learning. This provides the extension of the learning equations that were seen in section 5.1. For instance, the change in associative strength for CS_A at time *t* would be:

$$\Delta V_{A(t)} = \alpha_A \beta \left(\lambda - \Sigma V_{(t)} \right) \tag{16}$$

Note that this equation is sensitive to the fact that each CS might have a different salience; α_A is the salience for CS_A alone. Similar equations would be used to calculate the change in associative strengths for the other conditioned stimuli that were present, with each equation using a different salience constant.

5.2.2 Model Implications

The Rescorla-Wagner model provided an account of the varieties of associative learning that were introduced in Chapter 3. In particular, it made excellent predictions about learning situations that involved compound stimuli (for an introduction, see Pearce, 1997). In particular, it provided an explanation of blocking, as well as an account of overshadowing. Miller, Barnet, and Grahame (1995) have reviewed 18 different successes of the model.

This is not to say that the model is not without its problems. There have been 23 specific failures of the model that have been documented (Miller et al., 1995). These failures have led to the development of newer models (see Pearce, 1997, for an introduction to some of these). While some of these models have abandoned the general assumptions of the Rescorla-Wagner model (e.g., Gibbon, 1977; Miller & Matzel, 1988), others retain its general, but add variations that address some of its assumptions that are deemed to be problematic.

However, these newer models also have their own shortcomings, and are also more complicated than the original Rescorla-Wagner formulation. “For the time being, researchers would be well advised to continue using aspects of the Rescorla-Wagner model, along with those of other contemporary models, to help them design certain classes of experiments” (Miller et al., 1995, p. 381). The Rescorla-Wagner model is enduring and has been extremely influential.

As noted, the Rescorla-Wagner model provides an account of blocking and overshadowing. We saw in Chapter 3 that perceptrons can produce these phenomena as well. Furthermore, the Rescorla-Wagner equation is structurally very similar to the delta rule equation that was presented in Section 2.9. Are these parallels coincidental, or is there a deeper formal relationship between the delta rule and the Rescorla-Wagner model?

5.3 Formalizing Learning and Extinction

5.3.1 Changing λ

One of the purposes of the Rescorla-Wagner model is to formally describe how the repeated pairing of a CS with a US will alter the strength of the association between them. As we saw earlier, this can be formalized with the following iterative equation:

$$\begin{aligned} V_{(t+1)} &= V_{(t)} + \Delta V_{(t)} \\ &= V_{(t)} + \alpha\beta\left(\lambda - V_{(t)}\right) \end{aligned} \tag{17}$$

However, another purpose of the Rescorla-Wagner model is to formally describe how an existing association between the CS and US will weaken when the CS is presented in the absence of the US. This type of "learning" is called extinction.

How can the Rescorla-Wagner model be extended to handle extinction? The answer to this question is to change the value of λ on trials for which the animal is not reinforced (that is, on trials in which the US is not presented). Recall that λ "refers to the magnitude of the US, and a value of 0 is appropriate to indicate its absence" (Pearce, 1997, p. 58). In other words, we can use the equation given above to model extinction, provided we assign λ the value of 0.

Figure 5-1 illustrates the power of this one equation to model learning and extinction. It illustrates 100 trials of learning involving a single CS with the salience (α) of 0.35. (It is assumed that β = 1.) The initial association between the CS and the US was equal to 0. For the first 50 trials that make up the graph, the association was increased by using the Rescorla-Wagner model with λ equal 100. You will note that the value of V rapidly increases to this value after about 25 trials, and remains at λ until trial 50.

At trial 50, extinction begins by setting λ to a value of 0 in the Rescorla-Wagner model. You will note that the associative strength rapidly decreases to this value, and remains there until trial 100, at which time the simulated learning experiment ends.

5.3.2 Implications for the Perceptron

The importance of this demonstration is to highlight the fact that the values used in the Rescorla-Wagner model change as a function of experimental context. That is, λ has one value when an agent is being reinforced, and a different value when the agent is not being reinforced. This fact is all that is needed to translate the kind of learning described by the Rescorla-Wagner model into the kind of learning that is involved when a perceptron is trained.

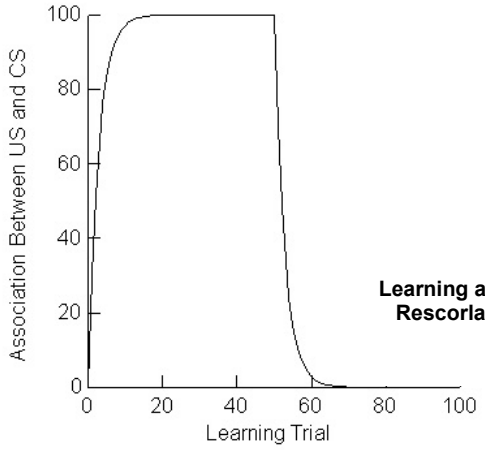

Figure 5-1.
Learning and extinction according to the Rescorla-Wagner model. See text for details

5.4 A Unified Rescorla-Wagner Equation

5.4.1 Three Learning Equations

In terms of use, there are three different versions of the Rescorla-Wagner equation that are used by researchers when they are modeling associative learning. These different versions of the equation depend upon the context of a particular experimental trial. That is, they depend on whether some US is present or absent, and upon whether a CS is present or absent.

The situation that leads to the first equation is the one in which both the US and some CS_A are present during a trial. In this case, the change in association for CS_A is modeled as follows:

$$\Delta V_{A(t)} = \alpha_A \beta \left(\lambda - \Sigma V_{(t)} \right) \quad (18)$$

The situation that leads to the second equation occurs when CS_A is present during a trial, but the US is not. We saw in Section 5.3 that the equation for this situation involves replacing λ with 0 as in the equation below:

$$\Delta V_{A(t)} = \alpha_A \beta \left(0 - \Sigma V_{(t)} \right) \quad (19)$$

The third situation is one in which CS_A is not presented at all. In this case, whether the US is present or not, the Rescorla-Wagner model assumes that the absence of CS_A means that its association will not change. Mathematically, this means that when CS_A is not present that:

$$\Delta V_{A(t)} = 0 \quad (20)$$

5.4.2 One Rescorla-Wagner Equation

The three different equations presented above are special, context-specific, cases of the general Rescorla-Wagner model. When relating our model to the network learning rules, it will be to our advantage to use a single expression that captures these three cases. This is because all of the perceptron learning that we have been examining is defined by a single expression.

To develop such an equation, let us define two variables. Let the variable US be equal to 1 when the unconditioned stimulus is present, and let it be equal to 0 when the unconditioned stimulus is absent. Similarly, let CS_A be equal to 1 when conditioned stimulus A is present, and be equal to 0 when this stimulus is absent. These two variables can now be used to write the Rescorla-Wagner model for learning involving CS_A:

$$\Delta V_{A(t)} = \alpha_A \beta \left(\lambda \cdot US - \Sigma V_{(t)} \right) \cdot CS_A \quad (21)$$

This equation will generate the three special case equations from Section 5.4.1. For instance, when the US and CS_A are both present, it produces the following:

$$\begin{aligned} \Delta V_{A(t)} &= \alpha_A \beta \left(\lambda \cdot US - \Sigma V_{(t)} \right) \cdot CS_A \\ &= \alpha_A \beta \left(\lambda \cdot 1 - \Sigma V_{(t)} \right) \cdot 1 \\ &= \alpha_A \beta \left(\lambda - \Sigma V_{(t)} \right) \end{aligned} \quad (22)$$

Similarly, when the US is absent and CS_A is present, the equation produces the following:

$$\begin{aligned} \Delta V_{A(t)} &= \alpha_A \beta \left(\lambda \cdot US - \Sigma V_{(t)} \right) \cdot CS_A \\ &= \alpha_A \beta \left(\lambda \cdot 0 - \Sigma V_{(t)} \right) \cdot 1 \\ &= \alpha_A \beta \left(0 - \Sigma V_{(t)} \right) \end{aligned} \quad (23)$$

Finally, when the US is present and CS_A is absent, the equation produces the following:

$$\begin{aligned} \Delta V_{A(t)} &= \alpha_A \beta \left(\lambda \cdot US - \Sigma V_{(t)} \right) \cdot CS_A \\ &= \alpha_A \beta \left(\lambda \cdot 1 - \Sigma V_{(t)} \right) \cdot 0 \\ &= 0 \end{aligned} \quad (24)$$

Of course, the equation will also produce a 0 change in the association if both the US and the CS are absent. Later, we will determine whether this unified equation can be translated into a learning rule for a perceptron.

5.5 A Picture of Similarity

A perceptron, trained with the delta rule, provides one account of associative learning. It is capable of modeling overshadowing and blocking, which both provided challenges to earlier theories of association. The main feature of this model is that association changes as a function of a difference between two elements: the desired response of an output unit and the observe response of the output unit when a stimulus is presented.

The Rescorla-Wagner model provides another account of associative learning, and is also capable of handling overshadowing and blocking. The main feature of this model is that association changes as a function of a difference between to elements: the maximum possible associative strength, and the current associative strength.

Given these similarities, it is natural to think that there might be a very powerful relationship between the two approaches to learning. Indeed, formal equivalence between the two models was first noted in the early 1980s (Sutton & Barto, 1981). That is, delta rule learning and Rescorla-Wagner learning are identical – under a very particular assumption about a network's activation function.

One approach to establishing the formal equivalence is based on the making the value of λ depend on whether or not a learning trial involves reinforcement. This permits us to create a new version of the perceptron learning diagram that was presented earlier as Figure 2-2. In particular, we can re-label that figure to bring its elements in line with Rescorla-Wagner model.

This is accomplished in Figure 5-2. Figure 5-2(A) is the same as Figure 2-2. Figure 5-2(B) presents the same figure, labeled in accordance with the Rescorla-Wagner model. There are two main differences that arise because of the re-labeling.

First, the connection weights in the perceptron are now shown to represent the strength of associations between the CSs and the US. Second, the value of the US is altered to move the delta rule into the Rescorla-Wagner formalization. In the delta rule, when the US is present it is given a value of 1 before being presented to the "Compare" operation; when the US is absent it is given a value of 0. In the re-labeling, when the US is absent it is given a value of 0, but when it is present it is now given the value of λ. The point of Figure 5-2(B) is this: if this perceptron is trained with the delta rule, but using the values that are indicated in the diagram, then the delta rule learning will be equivalent to Rescorla-Wagner learning.

Interestingly, this diagram is slightly misleading. To formally translate the one model into the other, we must make some assumptions concerning the activation function that is in the output unit. However, the reworking of the equations will lead to the same conclusion that the figure below alludes to: the delta rule and Rescorla-Wagner learning are formally equivalent.

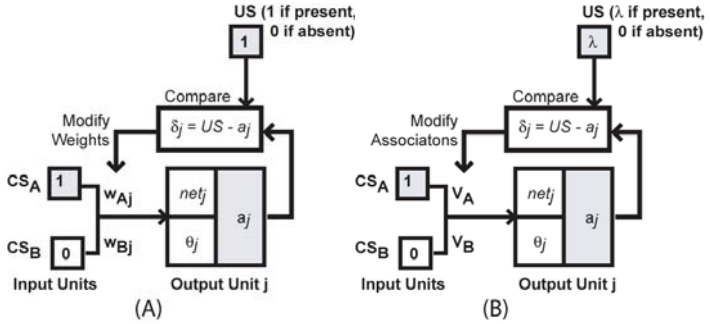

Figure 5-2. See text for explanation. Both figures illustrate a trial in which US and CS_A are present.

5.6 Formal Equivalence with Linearity

5.6.1 Activation Is Not Association

Why does Figure 5-2 only provide a hint of the equivalence between the Rescorla-Wagner model and the delta rule? The answer to this question can be found by examining the "Compare" box in the figure. As we have seen, the Rescorla-Wagner model works by comparing the maximum possible associative strength with the current associative strength. The re-labeling of Figure 5-2B incorporates the maximum possible associative strength into the delta rule by setting the value of US to either 0 or λ. However, it does not incorporate the current associative strength into the delta rule. As is shown in the "Compare" box, the maximum associative strength is compared to the activation of the output unit, not to the current associative strength.

5.6.2 A Linear Solution

In order to formally equate the two models of learning, we must find a way to translate the activity of an output unit (a_j) into the current total associations involving the CSs that are being presented to the learner (ΣV). Recall that in Figure 5-2B the current association of each CS to the US is represented as a connection weight. The total association in the system must be a sum involving these weights. One component of a perceptron in which such a sum can be found is the net input, which was discussed in Section 2.2.2. The net input is then converted into the perceptron's activity using the activation function. To equate the two learning models, a relationship between net input and total association must be specified, and must then be preserved in some way by the activation function so that output unit activity in the "Compare" box reflects total association.

In the traditional account of a perceptron, the net input is the weighted sum of the input unit activities. The activity of each input unit *i*, where the activity is expressed as a_i, is weighted by the strength of the connection between the input unit and the output unit *j* (w_{ij}). Thus, the net input for output unit *j* can be expressed as:

$$net_j = \sum_i a_i w_{ij} \qquad (25)$$

Gluck (Gluck & Bower, 1988; Gluck & Myers, 2001) noted that if the input units of a perceptron indicate the presence of a CS, and the connection weights represent the set of CS-US associations, then the net input equation given above will also compute overall associative strength. Recall from Section 5.2.1 that for a three CS example, overall associative strength was defined as:

$$\Sigma V = V_A + V_B + V_C \qquad (26)$$

To generalize this equation, let a_i be 1 if CS_i is present, and let it be 0 if CS_i is absent. Let V_i be the associative strength between CS_i and the US. Overall associative strength now becomes:

$$\Sigma V = \sum_i a_i V_i \qquad (27)$$

However, V_i in the above equation is identical to w_{ij}, which means that ΣV is identical to net input:

$$\Sigma V = \sum_i a_i V_i = \sum_i a_i w_{ij} = net_j \qquad (28)$$

In order to include ΣV (or net input) directly into the "Compare" box of Figure 4-2B, a linear assumption about the activation function of the output unit is made (Gluck & Bower, 1988; Gluck & Myers, 2001). That is, it is assumed that the activation function is the identity function, which means that output unit activity is identical to net input:

$$a_j = \sum_i a_i w_{ij} = net_j \qquad (29)$$

With this assumption about the activation function, the calculation of error in the delta rule is identical to the calculation of the "associative difference" in the Rescorla-Wagner rule. This is one example of a proof that the two learning rules are identical.

5.7 Nonlinear Activation and Overall Association

5.7.1 Nonlinear Activity

The preceding proof of the identity between the delta rule and the Rescorla-Wagner model required a linear relationship between an output unit's net input and its activity. Unfortunately, this requirement converts the perceptron into a much weaker type of artificial neural network, a distributed associative memory (e.g., Dawson, 2004, Chapter 9). Much of what makes a perceptron interesting depends on a nonlinear relationship between net input and activation.

As we saw earlier, early research on perceptrons used the Heaviside equation to enforce this nonlinear relationship:

$$a_j = H(net_j) = \begin{cases} 1 : net_j > \theta_j \\ 0 : net_j \leq \theta_j \end{cases} \quad (30)$$

This equation converts net input into either the value 1 or 0.

However, we have also seen that more modern treatments of the perceptron use other nonlinear equations which convert net input into a continuous range between 0 and 1. In particular, an integration device is a traditional perceptron in which the activation function of the output unit is the logistic function instead of the Heaviside equation.

Another nonlinear activation function that can be used in the perceptron is the Gaussian (e.g., Dawson, 2004; Yaremchuk, Willson, Spetch & Dawson, 2005). A Gaussian that ranges between 0 and 1, and which has a standard deviation of 1, is defined by the following equation:

$$a_j = G(net_j) = e^{(-\pi(net_j - \mu_j)^2)} \quad (31)$$

The Gaussian has a bell shape, asymptotes to 0 in both directions, and reaches a maximum value of 1 when the net input is equal to the mean of the Gaussian μ_j as is shown in Figure 5-4.

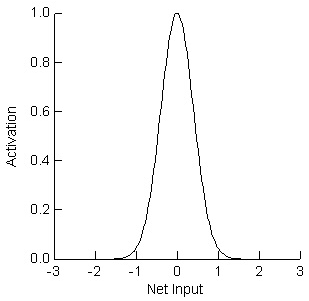

Figure 5-3. The Gaussian equation produces a bell-shaped activation function.

5.7.2 Nonlinear Association

We need to find a method by which an output unit's nonlinear activity can be converted into a measure of associative strength. All of the activation functions mentioned above range between 0 and 1, and that all are based on net input – which we have shown to be equal to overall association (ΣV). Therefore one plausible approach would be to use output unit activity to define overall association as a current proportion of the maximum possible association λ.

That is, when the connection weights adopt values that reflect that the maximum possible association has been achieved, activation would reach 1, and overall activation would be λ times 1. When no association is encoded in connection weights, activation would be 0, and overall association would be λ times 0. In short, let us propose that for a perceptron that uses some nonlinear activation function $f(net_j)$, overall association can be formalized as:

$$\Sigma V = \lambda \cdot f(net_j) \quad (32)$$

5.8 Formal Equivalence with Nonlinearity

In section 5.6.2, we defined overall association in terms of the nonlinear activity that is typically found in the output unit of a perceptron. We are now in position to demonstrate the equivalence between delta rule learning in this type of perceptron and Rescorla-Wagner learning.

Consider a variable US, which is equal to 1 on trials in which an animal is reinforced, and which is equal to 0 on trials in which an animal is not reinforced. Similarly, consider the variable CS_i which is equal to 1 when stimulus *i* is present, and which is equal to 0 when it is not. With these two variables we can define, for some time t, the change in the association between CS_i and the US with the following version of the Rescorla-Wagner model:

$$\Delta V_{i(t)} = \alpha_i \beta \left(\lambda \cdot US - \Sigma V_{(t)}\right) \cdot CS_i \qquad (33)$$

In Section 5.6.2 we provided a definition of overall association for a perceptron that uses a nonlinear activation function. We also noted in our discussion of Figure 4-2B that the value of CS_i is represented in a perceptron by the activation of input unit *i* (that is, by a_i). If we replace ΣV in the above equation with this definition, we obtain the following equation:

$$\Delta V_{i(t)} = \alpha_i \beta \left(\lambda \cdot US - \lambda \cdot f(net_j)_{(t)}\right) \cdot a_i \qquad (34)$$

Not that in this equation, λ is on both sides of the subtraction. So, this equation can be simplified by moving λ outside of the parentheses. As well, the value $f(net_j)$ is identical to the activity of the output unit, a_j. This permits the expression above to be rewritten as:

$$\begin{aligned}\Delta V_{i(t)} &= \alpha_i \cdot \beta \cdot \lambda \cdot \left(US - f(net_j)_{(t)}\right) \cdot a_i \\ &= \alpha_i \cdot \beta \cdot \lambda \cdot \left(US - a_{j_{(t)}}\right) \cdot a_i\end{aligned} \qquad (35)$$

Finally, we can note that α_i, β, and λ in the above equation are constants. We could replace the product of these three constants with another constant, η, without losing any generality. The result is the expression below:

$$\Delta V_{i(t)} = \eta \cdot \left(US - a_{j_{(t)}}\right) \cdot a_i \qquad (36)$$

The above equation represents a translation of the Rescorla-Wagner model into a situation in which the learning agent is a perceptron with a nonlinear activation function. There are two key points to note about it. First, this equation involves the direct comparison between US and a_j. This is exactly the difference that is required in the "Compare" box in Figure 4-2B. Second, the equation above is identical to the equation for the delta rule that was provided in Section 2.9.2, and which is given once again below:

$$\begin{aligned}\Delta w_{ij} &= \eta \cdot \delta_j \cdot a_i \\ &= \eta \cdot (t_j - a_j) \cdot a_i\end{aligned} \qquad (37)$$

Note that the identity of the Rescorla-Wagner model and the delta rule depends on our recognizing the relationships between learning values and perceptron components that was established in Figure 4-2B. In particular, a change in an association is identical to a change in a connection weight, and the desired value for the output unit, t_j, is identical to the US variable that we included in the Rescorla-Wagner equations on the current page.

In summary, we saw in Section 5.5 that if the delta rule is used to train a perceptron with a linear activation function, then the delta rule in this case is equivalent to the Rescorla-Wagner model. In this section, we have generalized this statement by proving that the delta rule is equivalent to Rescorla-Wagner learning when the perceptron has a nonlinear activation function that ranges between 0 and 1. This proof depends upon the assumption that the activation value of a perceptron's output unit can be interpreted as representing the proportion of the maximum association that is currently encoded in the connection weights.

5.9 The Implications of Equivalence

A perceptron trained with the delta rule, and an associative system trained with the Rescorla-Wagner equation, have been shown earlier to both be capable of providing accounts of complex phenomena such as overshadowing and blocking. In this chapter, we have seen that the similarities between the two should not be surprising – by labeling a perceptron's components in a particular way, the delta rule for training it can be directly translated into the Rescorla-Wagner rule.

Establishing such a relationship between animal learning and perceptron learning is important, because we have an excellent formal understanding of what perceptrons can learn, and cannot learn, to do (Minsky & Papert, 1988). As a result, it may be possible to translate this knowledge from the domain of machine learning into the domain of animal learning.

For example, a variation of a perceptron was proposed for solving a number of interesting categorization problems (Gluck, 1991). However, Pearce (1997, p. 132) critiqued this model, noting "because it is formally equivalent to the Rescorla-Wagner model, it makes the same erroneous predictions as that model". In other words, problems that can be learned by a perceptron should also be explainable by the Rescorla-Wagner model. Problems that cannot be learned by a perceptron should also pose problems for the Rescorla-Wagner rule.

The problem with this perspective is that all that has been established is the equivalence between the two learning rules. It is important not to read too many additional similarities into the relationship between perceptrons and animal learners.

In particular, the Rescorla-Wagner model *only* provides an account of how associations change over time. It is completely mute with respect to how associations are converted into observable responses. To place this observation into connectionist terms, the Rescorla-Wagner model describes how associations can be modified to produce net input. The model does not describe how net input is converted into activation, because that isn't part of learning.

In contrast, perceptrons describe how existing connection weights are translated into observable responses. This was the reason that we had to devote effort in Sections 5.6 and 5.7 to define the relationship between overall association and activation.

We will see that this becomes an issue when we consider the role of the activation function in the next chapter. For example, we will explore claims that certain learning paradigms cannot possibly be accounted for by a perceptron trained with the delta rule. We will see that this is the case for one perceptron that uses one nonlinear activation function. However, if the activation function is replaced with another, we will see that the perceptron can model the learning. The problematic issue, of course, is that both perceptrons are being trained with the same learning rule (i.e., the general rule described in Section 5.7). Clearly there are differences between network and animal learning that are not captured by the formal equivalence of the learning rule.

In a similar vein, if one uses the equivalence between learning rules to claim that perceptrons are equivalent to Rescorla-Wagner models of animal learning, then one would expect that a phenomena that can be explained by the Rescorla-Wagner model would also be modeled by a perceptron. We will see that this is not necessarily true. The correspondence between the two learning theories is strongly influenced by the activation function that is part of the perceptron's output unit, and by the fact that in a perceptron learning requires that such activation must be produced.

Chapter 6: Context, Bias, and the Null Condition

This chapter examines some design decisions that must be made when artificial neural networks are used to simulate animal learning experiments. These design decisions have an impact upon the relationship between machine learning and animal learning models. Three such decisions are explored: the relationship of a network's bias to the influence of uncontrolled background stimuli, the empirical and logical need to include a null condition in which a network is trained to turn off when no CSs are presented, and the ability to replace a monotonic activation function with a nonmonotonic activation function. Considering these issues highlights subtle relationships between the perceptron and classical conditioning. However, at the end of the chapter we will also see that this consideration introduces a puzzling issue: the responses of a perceptron is not necessarily identical to those predicted by the Rescorla-Wagner model.

A peer-reviewed monograph published by Comparative Cognition and Behavior Reviews on behalf of the Comparative Cognition Society.

6.1 Design Decisions

6.1.1 The Journey Thus Far

At this point, we have identified two general points of contact between a particular version of network learning (the delta rule) and a particular model of animal learning (the Rescorla-Wagner model). The first has been empirical: we have shown that perceptrons can produce many results that are analogous to those found in the animal learning literature. The second has been formal: we have shown that one can translate the Rescorla-Wagner equation into the delta rule for the perceptron, even when that rule is applied to a perceptron with a nonlinear activation function.

These two points of contact illustrate a comfortable relationship between the two different approaches to learning. This type of relationship has existed ever since Sutton and Barto (1981) first described the similarities between the two types of learning rules. "The perceptron is essentially the same as the Rescorla-Wagner model: it compares its own response (expectation) with the correct one (US) and modifies the weights in order to make them agree" (p. 156).

Our purpose is to provide a more careful examination of the relationship between these two views of learning. Soon we will be in a position to consider some thought-provoking examples that challenge it. However, before we can do this, some subtler points of contact between network learning and animal learning must be considered. These considerations are provided in the current chapter.

6.1.2 Design Decisions to Consider

When we simulate an animal learning experiment by training a perceptron, the perceptron can be viewed as a particular operationalization of a theory (e.g., a theory of animal learning like the Rescorla-Wagner model). One property of a working computer simulation is that it leaves no room for vagueness: to make the simulation work, the theory must be rigorously specified (Lewandowsky, 1993). One aspect of this rigorous specification is the making of design decisions. A design decision occurs when a researcher translates a component of a theory into a working part of a simulation. Design decisions are critical because, in many instances, there is more than one way to translate the theory into a working model.

For example, one design decision that is important to the studies of learning in this book concerns the activation function to insert into the output unit of the perceptron. We have already established two different candidate activation functions: the Heaviside equation in a standard perceptron, and the logistic equation in an integration device. Later in Chapter 6 we will consider the implications of choosing a third activation function, the Gaussian.

A second design decision that is important to simulations that are related to the Rescorla-Wagner model concerns how the effects of context are to be modeled. One of the assumptions of the Rescorla-Wagner model is that there exists a context, not under direct experimental control, that is present on every learning trial and which can be conditioned as if it were another CS. When training an integration device, this type of context can be modeled "for free" – that is, without specifying additional input units. This issue will also be discussed in this chapter.

A third design decision that has an impact on simulations of conditioning concerns situations in which neither the US nor any CSs are presented. What should happen to the perceptron in this situation? Should a "null condition" be included when a perceptron is trained, or not? These questions are important, and will also be answered in this chapter.

6.2 Context and Bias

6.2.1 The Need for Context

Classical conditioning does not occur in a vacuum. For instance, the Rescorla-Wagner model makes explicit the assumption that conditioning is context-dependent. "The changes in associative strength of a stimulus as a result of a trial can be well-predicted from the composite strength resulting from all stimuli present on that trial" (Rescorla & Wagner, 1972, p. 73). In Chapter 4, we saw several examples that supported this claim. For example, the ease with which some CS_B can be conditioned depends upon whether or not it is presented with some other CS_A that is a conditioned inhibitor (e.g. Section 4.4).

6.2.2 Background Stimuli

In any simulations in which context was presumed to be explicitly present (e.g., Sections 4.4., 4.6, 4.8), context was under direct experimental control. The context for one CS was determined by presenting other CSs that were represented by their own input units. This is analogous to a paradigm in which an experimenter controls or manipulates particular aspects of a context for an animal's learning (e.g., control of lights, odor, texture, etc).

Rescorla and Wagner (1972) recognized the importance of another kind of context: an array of background stimuli that were always present, and not under the experimenter's control. "The CS occurs against a background of uncontrolled stimuli. To speak of shocks occurring in the absence of the CS is to say that they occur in the presence of situational stimuli arising from the experimental environment. Although these stimuli are not explicitly manipulated by the experimenter, they nevertheless can be expected to influence the animal" (p. 88).

Rescorla and Wagner (1972) showed how their model could easily accommodate this background of uncontrolled stimuli. First, they assumed that this background was in essence a CS that was present on every trial. Second, they assumed that this CS could be treated as if it was any other CS – that is, it had an associative strength that changed as a function of the Rescorla-Wagner equation.

6.2.3 Bias as Background

This view of context is not universally accepted, but is an important component of the Rescorla-Wagner model. It is therefore critical to determine how the notion of a background of uncontrolled stimuli can be realized in an integration device.

In all of the simulations that have been presented in this book, a learning rule has not only been used to modify connection weights, but has also been used to modify the bias (θ) of the activation function. As was noted earlier in Section 3.1.2, bias is modified by assuming that θ is the weight of a connection between the output unit and an "extra input unit" that always has an activation of 1. The assumed existence of this extra input unit permits θ to be treated exactly like any other weight in the network.

This treatment of bias by the learning rule also permits it to be directly related to the Rescorla-Wagner formulation. The "extra input unit" that is always on represents a background CS that it always present. When θ is manipulated by the learning rule, this is equivalent to the associative strength of the background CS being modified in accordance with the Rescorla-Wagner equation. Thus all the simulations reported thus far in which θ has been modified have included a tacit assumption that there exists a contextual CS that has an associative strength that contributes to overall conditioning. This CS is always in addition to any controlled CSs that are explicitly manipulated as part of the context of a conditioning experiment. Bias is the associative strength of this background stimulus. This background stimulus is a conditioned excitor when θ is positive, and which is a conditioned inhibitor when θ is negative.

6.3 Negative Contingency As An Example

6.3.1 Negative Contingency

In Chapter 4, we observed the performance of networks trained in a particular paradigm (Pavlov, 1927) that was designed to convert one of the CSs into a conditioned inhibitor. An alternative approach for doing this is called the negative contingency paradigm (e.g., Rescorla, 1969). In its simplest form, the negative contingency paradigm is designed to convert some CS_A into a conditioned inhibitor. This is accomplished by employing two different training trials. When the US is presented, CS_A is absent. When CS_A is present, the US is not. CS_A becomes a conditioned inhibitor because it is recognized as a signal that the US will not occur. Importantly, this paradigm assumes that the experimental context serves as a conditioned excitor that signals the presence of the US. The purpose of this section is to demonstrate this in an integration device.

6.3.2 Retardation of Acquisition Test

All networks were trained on two patterns in a pretraining (negative contingency, see Table 6-1) phase until the networks converged. Without reinitializing weights, control networks were then trained to respond to CS_B (i.e., B+ conditioning). In contrast, experimental networks were then trained to respond to a compound of CS_A and CS_B (i.e., AB+ conditioning). If CS_A is a conditioned inhibitor, then experimental networks should require more training than control networks.

Phase	Trial Type	CS_A	CS_B	t_j
Pretrain	+	0	0	1
	A-	1	0	0
Control	B+	0	1	1
Experimental	AB+	1	1	1

Table 6-1

This is exactly what was found. 25 different networks served as "subjects" in each condition. Control networks converged on average after 3.40 sweeps, while experimental networks converged on average after 60.96 sweeps. This difference was statistically significant (t = -87.615, df = 48, p < 0.0001), and indicates that CS_A served as a conditioned inhibitor.

6.3.3 Effect on Bias

How does the negative contingency paradigm affect contextual conditioning? Table 6-2 provides the connection weights from the two input units, and the bias of the logistic activation function in the output unit, for a typical network prior to any training taking place. Note that the two connection weights have very small values, and that the bias is equal to zero. This structure is the result of initializing the network using the standard practices that were described earlier.

Training	Unit	Weight	θ
Before Phase 1 Learning	CS_A	0.01	-
	CS_B	0.02	-
	Output	-	0.00
After Phase1 Learning	CS_A	-4.63	-
	CS_B	0.02	-
	Output	-	2.20

Table 6-2

Table 6-2 also provides the structure of the same network after negative contingency training (i.e., after pretraining with the Table 6-1 patterns). It reveals three important points. First, CS_A has developed a strong negative weight. This is exactly as we would expect if this stimulus has become a conditioned inhibitor. Second, the weight for CS_B has not changed. This too is expected, because this stimulus was absent during pretraining. Third, the bias has obtained a strong positive value.

This last result is because the integration device was trained to turn on when neither CS was present. How is this to be achieved? The bias must be big enough to produce logistic activity of 0.9 or higher when no other CSs are present. Section 6.2 proposed that bias reflects the strength of the experimental context's association. The positive bias in Table 6-2 reveals that the context has become a conditioned excitor, which is a plausible consequence of the negative contingency paradigm.

6.4 Bias, Context, and Inhibition

6.4.1 Context and Excitation

In Section 6.2 it was argued that bias of could represent the associations involving background context. In many (but not all) of the simulations reported in this manuscript, the bias in an output unit at the end of training is negative. The natural interpretation of this is that training has resulted in the background context becoming inhibitory.

However, many experiments have demonstrated that the result of training is to produce contexts that are excitatory, *not* inhibitory (e.g., Balsam, 1985; Bouton & King, 1983; Chang, Stout & Miller, 2004; Grahame, Barnet & Miller, 1992). This would appear to pose a problem for the network models that we have been considering.

6.4.2 Empirical Possibility of Inhibition

One response to this is that other studies have produce results that contradict those cited above (e.g., Durlach & Rescorla, 1983; Marlin, 1982; Stout & Miller, 2004), suggesting that it is premature to claim that excitatory contexts are obligatory.

Further to this, Schmajuk, Larrauri, and LaBar (2007) tested a neural network model of classical conditioning that incorporated attentional processes (Schmajuk, Lam, & Gray, 1996). As part of their study, Schmajuk, Larrauri and LaBar used this model to simulate an experiment whose results had previously been used to argue for inhibitory context (Bouton & King, 1983). Their simulation replicated the experimental results by acquiring inhibitory associations involving the context. However, it also revealed that reduced attention to the context would make it difficult to observe the inhibitory influence of context on animal behavior.

6.4.3 Formal Issues

Some formal considerations also enter into the deliberation of context and inhibition. First, the notion of whether bias is positive or negative depends completely upon input encoding. For example, consider a network that uses the Heaviside activation function, and codes the absence of conditioned stimuli with 0s, and their presence with 1s. At the end of Pavlov's (1927) conditioned inhibition training (A+, AX-), the network will have a bias of -2.36, and will assign a weight of 4.56 to CS_A and a weight of -4.80 to CS_X.

Now, imagine encoding the absence of conditioned stimuli with 5s, and their presence with 6s. A logically equivalent network – one that generates exactly the same responses to the same stimuli (with different input codes) – has exactly the same weights as the previous network, but now has a bias of 1.64. In other words, it is not necessary to have negative biases, because one can translate one network into an equivalent one that has a different sign of bias. The translation is accomplished by varying the input code, which, of course, is arbitrary.

This highlights a disconnect between empirical and simulation studies, at least with respect to inhibition. On the one hand, theories like the comparator hypothesis assume all associations are excitatory, and have impressively handled a great deal of empirical results (e.g., Miller & Matzel, 1988).

On the other hand, inhibition is central to simulation studies of associative learning. For instance, simulations revealed that Hebb's (1949) theory of the creation of cell assemblies – a theory central to the formation of associations in the comparator hypothesis (e.g., Savastano et al., 2003, p. 392) – required inhibition in order to function properly (Milner, 1957; Rochester et al., 1956).

Clearly one contribution of the simulations discussed in this manuscript is to raise the issue of context and inhibition. Should a model be rejected if it requires inhibitory context? Is it possible that empirical studies will have difficulty identifying such inhibitory effects? Importantly, such questions are raised as a result of attempting to simulate associative learning, and attempting to carefully relate properties of networks (such as bias) to associative learning concepts.

6.5 Defining the Null Condition

6.5.1 The Need for Nothing

In the preceding section, one of the training patterns simulated the situation in which the US was presented, but the CS was not. We could designate this condition with the symbol +, because the learning system is being reinforced in the absence of conditioned stimuli. In this section, we consider the need for an alternative situation, which we will call the null condition, and which we will represent with the symbol -. In the null condition, the learning system is not presented any stimuli (US or CS). That is, the system is *not* reinforced when no CSs are present.

One reason that we might be interested in including the null condition in our training of artificial neural networks is because it is an important component of animal learning experiments. During such an experiment, USs are not continually present. A US will be presented for a brief period of time, and then there will be an interval of time during which the US is absent. The time between US presentations is called the intertrial interval (ITI) or the cycle duration (Kirkpatrick, 2002). It is important because variations in the ITI can affect conditioning. For example, increasing the duration of the ITI produces increases in the degree of conditioned taste aversion (Domjan, 1980).

A second reason for including the null condition is that it is tacitly assumed to exist in the Rescorla-Wagner model. Recall that the central assumption of this model is that a CS is a predictor of a US's occurrence. When conditioning produces a strong positive association for a CS, this means that the presence of the CS is an excellent predictor that the US is going to occur. However, this must also mean that when this CS is *not* present, then its absence is an excellent predictor that the US is *not* going to occur. This latter interpretation is identical to assuming that the null condition must exist.

Alternatively, we can say that theories of learning are tacitly discriminative. When the association to CS_A is modified by A+ conditioning, there is a tacit assumption that this association is intended to discriminate A+ situations from – situations.

A third reason for including a null condition is that the role of the CSs may be to signal to presence of the US, and that the role of context – that is, background stimuli – is to signal the absence of the US. In short, in many cases context should become a conditioned inhibitor. To achieve this, we need to include a condition in which no controlled CSs are present, and in which the US is also absent.

A fourth reason for including a null condition will be illustrated in the following pages. In many cases, when the null condition is included in a simulation, the results of the simulation are more interesting, accurate, or compelling. For example, we will see that the null condition facilitates overshadowing and blocking in the integration device.

A fifth reason for including a null condition is that it is required for computational claims about associative learning to be correct (Yaremchuk, Willson, Spetch & Dawson, 2005). It is often argued that one paradigm in animal learning, called negative patterning, is identical to one type of machine learning problem, XOR. However, this identity requires that negative patterning includes the null condition. Simulations of negative patterning must include the null condition to maintain the logical relationship between animal learning and network learning. Importantly, some simulations of negative patterning do *not* include the null condition (e.g., Delamater, Sosa & Koch, 1999), and are therefore studying a problem that is much simpler than XOR.

6.5.2 The Null Condition

The null condition in a network simulation, then, is simply a pattern in which all input units have activities of 0, and the output unit is trained to turn "off". Let us now see how the inclusion of this pattern can affect some of the results that were originally reported in Chapter 4.

6.6 Overshadowing With A Null Condition

6.6.1 An Alternative Training Set

Overshadowing in the integration device was examined earlier in Section 4.9. In that study, CS_A was ten times less intense than CS_B. Simulation results revealed a statistically significant overshadowing effect. That is, after AB+ training, the connection weight from CS_A was substantially smaller than the connection weight from CS_B. Furthermore, when CS_A was presented alone to the trained network, it produced a statistically smaller response than was the case when CS_B was presented to the network.

However, one concern with that earlier study was that the magnitude of the effect that was observed was not particularly compelling. The response that the network generated to CS_B was 0.90, while the response to CS_A was 0.75.

One problem with that earlier simulation was that it did not include the null condition. To correct this situation, the simulation was carried out with a training set that included two stimuli: the null condition (i.e., trial type -), and the compound of the two stimuli (AB+). The training set used in this version of the overshadowing study is presented in Table 6-3:

Type	CS_A	CS_B	t_j
-	0	0	0
AB+	0.1	1	1

Table 6-3

6.6.2 Results

The training set in Table 6-3 was used to train 25 different perceptrons as subjects. After this training, the average response to the individual presentation of CS_B was 0.90, while the average response to the individual presentation of CS_A was 0.10. This difference was statistically significant ($t = 2064.689$, $df = 48$, $p < 0.0001$). Furthermore, this difference is far more dramatic than the one that was observed in Section 4.9, and provides much more compelling evidence for overshadowing in the integration device. Clearly the inclusion of the null condition can have a powerful influence on the simulation results.

The inclusion of the null condition does not dramatically alter the relationship between the weights from the two input units. Table 6-4 provides the structure of a typical network from this study. It can be seen that the weight associated with CS_A is an order of magnitude smaller than the weight associated with CS_B, which is consistent with the fact that there is an order of magnitude difference in the intensities of the two stimuli.

Unit	Weight	θ
CS_A	0.40	-
CS_B	4.60	-
Output	-	-2.20

Table 6-4

Why does the null condition enhance the overshadowing effect? Consider the responses of the integration device in terms of stimulus generalization. If we describe the input stimuli as vectors, then the network has learned to respond to [0.1, 1] and has learned to not to respond to [0, 0]. When the network is tested, it is presented a stimulus that it has not seen before – in vector terms, the stimulus [0.1, 0]. One advantage of artificial neural networks is that they generate similar responses to similar stimuli (McClelland, Rumelhart, & Hinton, 1986). If we measure the similarity of one vector to another by taking the Euclidean distance between them, then the stimulus [0.1, 0] is far closer to the original stimulus [0, 0] (distance = 0.01) than to the other original stimulus [0.1, 1] (distance = 1.0). Thus a weak response to this stimulus – consistent with overshadowing, and with the similar previous stimulus – is not surprising.

A complementary explanation is to note that the null condition has caused the context to be a strong conditioned inhibitor. As a result, when a weak CS is presented, the network will generate a much weaker response than when a strong CS is presented, because a weak CS will not be able to overcome the negative bias.

6.7 Blocking With A Null Condition

6.7.1 An Alternative Training Set

In Section 4.10, blocking in the integration device was studied. As was the case with overshadowing, a statistically significant blocking effect was found. However, the magnitude of the effect was not compelling. In this section, we repeat the blocking experiment, but add the null condition to the training sets.

The new training sets are provided in Table 6-5. 25 experimental networks were trained to convergence on the Phase 1 patterns (which required on average 421.2 sweeps). The networks were then given continued training on the Phase 2 patterns (convergence being achieved after only a single sweep). 25 control networks were trained on Phase 2 only; on average these networks converged after 281.0 sweeps.

Phase	Type	CS_A	CS_B	t_i
1	-	0	0	0
	A+	1	0	1
2	-	0	0	0
	AB+	1	1	1

Table 6-5

6.7.2 Results

Does prior training on CS_A block later training on CS_B? The standard test of blocking is to compare control responses to CS_B to experimental responses to the same stimulus. In this study a substantial blocking effect occurs. On average, the control networks generate a response of 0.55 to CS_B. In contrast, experimental networks generate an average response of 0.10 to CS_B – functionally equivalent to "off". This difference is statistically significant ($t = 160.974$, $df = 48$, $p < 0.0001$).

There are two major observations to make about these results in comparison to those that were reported earlier in Section 4.10. First, there is a much higher degree of blocking observed in this study. Second, the response of control networks to CS_B is much weaker in this study than the previous one (earlier, the control networks generated a response of 0.81 to this stimulus). Both of these effects are due to the effect of the null condition on the output unit's bias. This is revealed by studying the structure of the networks trained in this experiment.

Table 6-6 provides the structure of a typical experimental network after Phase 1 and after Phase 2 of training. From this, it can be seen that the first phase of training converted CS_A into a conditioned excitor (note the positive weight) and the context into a conditioned inhibitor (note the negative bias). CS_B was not involved in Phase 1 training, and therefore had a near zero weight. Note, though, that the network at the end of Phase 2 is nearly identical to the Phase 1 network. CS_B is blocked in the sense that its weight was unaltered by further training.

Phase	Unit	Weight	θ
Phase 1	CS_A	4.63	-
	CS_B	0.09	-
	Output	-	-2.20
Phase 2	CS_A	4.64	-
	CS_B	0.09	-
	Output	-	-2.20

Table 6-6

In comparison, Table 6-7 provides the structure of a typical network trained in the control condition (i.e., on Phase 2 only). Because there was no prior training on CS_A, there is no blocking, as revealed by the fact that both input units develop strong positive weights. That is, both CSs become conditioned excitors. Again, the null condition causes the context to become a conditioned inhibitor, as indicated by the strong negative bias. However, this is different than the networks described in Section 4.10, which developed positive biases. Because the bias is a strong negative in this study, the network only generates a moderate response when either CS_A or CS_B are presented alone

Unit	Weight	θ
CS_A	2.44	-
CS_B	2.37	-
Output	-	-2.20

Table 6-7

6.8 What Problems Can Perceptrons Solve?

6.8.1 Computational Analysis

The preceding two sections have demonstrated some empirical benefits to including the null condition as a training pattern in the network simulations. We now turn to a more abstract discussion of what the null condition purchases.

Earlier, we worked through the formal relationship between the delta rule and the Rescorla-Wagner model. One advantage of establishing this type of relationship is that knowledge that has been obtained in one domain, network learning, can now be carried into a different domain, animal learning. Researchers have conducted extensive analyses of perceptrons, and have a very rigorous understanding of what kinds of problems this type of network can solve (Minsky & Papert, 1988). Presumably there are many benefits to applying the results of such computational analyses to models of animal learning.

6.8.2 Linearly Separable Problems

What kinds of problems can perceptrons solve? One way to answer this question is to represent a to-be-learned problem in a pattern space. A pattern space is a spatial representation of all of the patterns that are to be presented to a network. Each pattern is represented as a point in this space. Each input unit in the network defines a dimension of the space. The coordinates of the point are determined by the activity that each input unit uses to represent the pattern.

Consider the logical operator AND that was discussed in Section 2.3. The training set required to teach AND to a perceptron is provided in Table 6-8.

Pattern	X	Y	t_i
1	0	0	0
2	1	0	0
3	0	1	0
4	1	1	1

Table 6-8

The pattern space for this problem requires that four different points be plotted in a two-dimensional space. The coordinates of each point are the X and Y values in Table 6-8. Figure 6-1 illustrates this space.

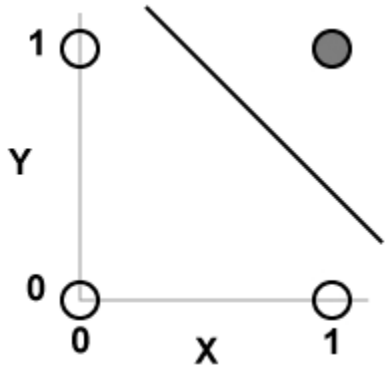

Figure 6-1.
Carving a pattern space to solve the AND problem.

There are two additional properties that are illustrated in Figure 6-1. First, each point is colored to indicate whether the network should turn on or off to the pattern. The three of the points that require an "off" response are white. The fourth point, to which the network must turn on, is grey.

Second, there is a line in the figure separating the grey point from the white points. When a single straight cut can be made in a pattern space to separate classes of patterns, then the problem represented by the pattern space is *linearly separable*. If a problem is linearly separable, then a perceptron can solve it.

One way to think of a linearly separable problem is as follows: if a problem is linearly separable, then there exists a single threshold such that the net input for each pattern in one class will be greater than the threshold, and the net input for each pattern in the other class will be less than the threshold. The output unit of a perceptron or an integration device provides this threshold. That is, each output unit of such a network makes a single straight cut in a pattern space.

6.9 What Problems Are Beyond A Perceptron?

6.9.1 Carving and Complexity

A linearly separable problem is the simplest kind of classification problem that a network can face. Problems that are more complex require a more complicated carving of the pattern space into different decision regions (Lippmann, 1989). Problems that require more than one straight cut through a pattern space are *linearly nonseparable*. Such problems cannot be solved by a perceptron or an integration device.

6.9.2 An Example Problem

One example of a small problem that is linearly nonseparable is another logical operator, XOR (for *exclusive or*). This operator, like AND, is defined over two input variables. If both variables are false, or if both variables are true, then XOR is false. XOR is only true if one input variable is true while the other input variable is false. If we were to attempt to train a perceptron on the XOR problem, then we would use the training set that is presented in Table 6-9:

Pattern	X	Y	t_j
1	0	0	0
2	1	0	1
3	0	1	1
4	1	1	0

Table 6-9

However, no matter how long we trained a perceptron or an integration device on the four patterns in Table 6-9, the network would never converge by generating a hit to every pattern. Instead, in the best situation, the network would generate a hit to three of the patterns, but would produce an incorrect response to the fourth. This is due to the linear nonseparability of XOR, which cannot be represented in the structure of a standard perceptron.

The pattern space for XOR is very similar to the one for AND that was illustrated in Section 6.7. The four patterns that are plotted in this space are identical to the four that were plotted in Figure 6-1. The difference between the two pattern spaces is the desired response to each pattern. Figure 6-2 provides the pattern space for XOR.

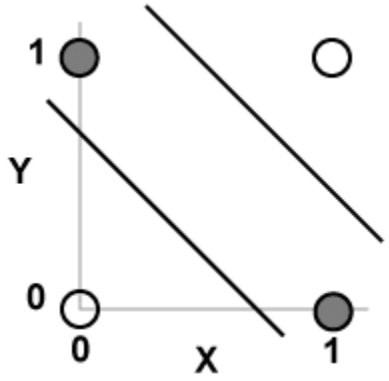

Figure 6-2.
Carving a pattern space to solve the XOR problem.

Note that in this pattern space the two middle points are drawn in grey because they require the perceptron to respond to them. To separate these two patterns from the other two white points, to which the perceptron should fail to respond, two straight cuts are required, as is illustrated in Figure 6-2. A single cut will fail to segregate all of the grey points from all of the white points. This is the reason that XOR is said to be linearly nonseparable.

Why is a traditional perceptron or an integration device unable to represent a solution to XOR? The reason is that a single threshold is not sufficient to use net input to separate all patterns of one type from all patterns of the other type. Instead, two thresholds (θ_1 and θ_2) are required. If the net input is lower than θ_1, then the network should turn off to the pattern. If the net input is higher than θ_2, then the network should also turn off to the pattern. If the net input is between θ_1 and θ_2, then the network should turn on to the pattern. Because the output unit of a standard perceptron or of an integration device only provides one threshold instead of two, these networks cannot represent a solution to this type of problem.

6.10 Negative Patterning and XOR

6.10.1 Negative Patterning

In negative patterning, a system is trained to respond when a component is presented alone (e.g., to respond to either CS_A or to CS_B), and is trained to not respond to CS_A and CS_B when they are presented together as a compound stimulus.

Negative patterning can easily be observed in studies of animal learning (Wagner, 2003). However, it cannot be explained by the Rescorla-Wagner model that was presented in Chapter 5 (Pearce, 1997). This is because in negative patterning, V_A and V_B must both be positive to generate strong responses to A and B. Therefore when AB is presented, the total associative strength $V_A + V_B$ must be even larger. Thus, the model predicts stronger responding to the compound – which is exactly the opposite of what is required.

6.10.2 Patterning and the Null Condition

Many researchers have argued that XOR is equivalent to negative patterning. This means that negative patterning is a linearly nonseparable problem that cannot be solved by a perceptron (Roitblat & von Fersen, 1992). Given that perceptron learning is identical to Rescorla-Wagner model, there should be no surprise in the latter's inability to deal with negative patterning, or with any other linearly nonseparable problem.

However, there is an important requirement in this argument (Yaremchuk et al., 2005). Consider a simple negative patterning paradigm; it would usually be described as A+, B+, and AB- (Delamater et al., 1999). The pattern space for these conditions is provided in Figure 6-3, and reveals a linearly separable problem that can easily be solved by a perceptron!

For this problem to be truly linearly nonseparable, the null condition *must* be included. Negative patterning must be defined as A+. B+, -, and AB-. The pattern space for this definition is provided in Figure 6-4.

That the definition of negative patterning must include the null condition might seem like nitpicking. However, it reveals a crucial design decision in simulation studies of negative patterning. If the null condition is not included, then negative patterning is not being simulated – that is, if it is assumed that negative patterning is identical to XOR. Delamater, Sosa and Katz (1999) simulated negative patterning but did not include the null condition. As a result, their networks were possibly learning a task that had a different logical structure than the one learned by their animal subjects. This might account for the discrepancies that they observed between their simulation and experimental data (Yaremchuk et al, 2005).

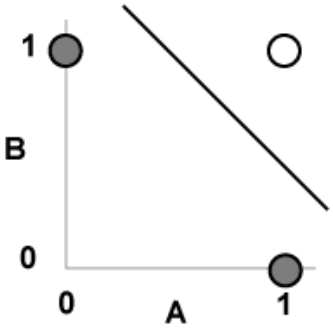

Figure 6-3.
The pattern space for the standard definition of negative patterning. It is linearly separable.

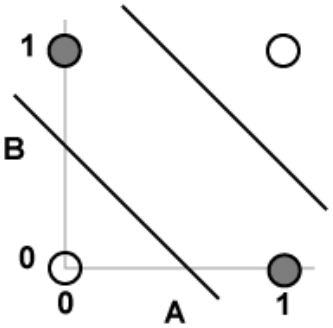

Figure 6-4.
Negative patterning with the null condition. It is linearly nonseparable.

6.11 The Value Unit

6.11.1 Nonmonotonic Activation

The fact that XOR and negative patterning are linearly nonseparable, and cannot be learned by an integration device, does not necessarily mean that such problems cannot be dealt with by perceptrons. The reason that an integration device cannot solve a problem like XOR is because its activation is monotonic: one can never decrease activity by increasing the net input that is passed into the activation function. As was discussed in Section 6.8.2, this means that the logistic equation behaves as if it has only one threshold, when two are required to solve XOR.

The typical solution to this problem is to model associative learning with more complicated artificial neural networks, networks that include at least one layer of processors between the input units and the output units. Such units are often called hidden units, and networks that include them are called multilayer perceptrons. "Application of this kind of model both solves knotty problems in animal learning and suggests a structure for the kinds of representations animals form during learning" (Roitblat & von Fersen, 1992, p. 678). Broad overviews of multilayer networks applied to animal learning are available (Enquist & Ghirlanda, 2005).

Importantly, this is not the only solution to this problem. Rather than adding hidden units to the architecture, one can deal with many linearly nonseparable problems in animal learning by adopting a different activation function in the output unit. In particular, the activation function can be nonmonotonic. This type of activation behaves as if it has two thresholds. Low activity is generated for net inputs that are lower than θ_1. High activity is produced for net inputs that are between θ_1 and θ_2. Low activity results when net inputs are θ_2. This relationship between net input and activity is nonmonotonic because increases in net input do not always produce increases in activity.

6.11.2 The Value Unit

One candidate nonmonotonic activation function is the Gaussian, which is illustrated in Figure 6-5, and which was briefly introduced in Section 5.7. The equation for the illustrated Gaussian is:

$$a_j = G(net_j) = e^{(-\pi(net_j - \mu_j)^2)} \quad (38)$$

This particular equation has a mean that is represented by the symbol μ, and a standard deviation equal to 1. This equation asymptotes to an activation of 0 when net input reaches positive or negative infinity. When the net input is equal to μ, the equation generates a maximum value of 1. As can be seen from Figure 6-5, this activation function is highly tuned – it only generates high activity to a narrow range of net input values. For this reason, an output unit that uses this activation function is called a value unit, using the terminology of Ballard (1986).

One way to view this activation function is as one that makes two parallel straight cuts through a pattern space. Because of this, a perceptron that uses this activation function can solve XOR or negative patterning without requiring additional hidden units.

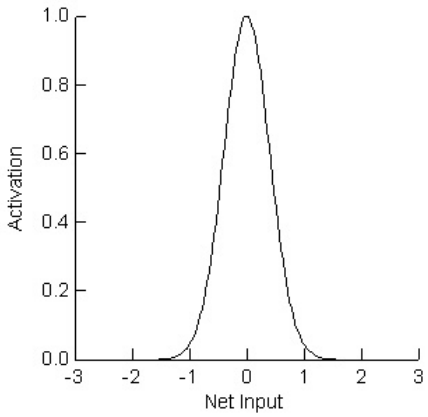

Figure 6-5.
The Gaussian activation function of a value unit.

6.12 A Learning Rule for a Value Unit

6.12.1 Error Minimization

The gradient descent rule that is used to train an integration device minimizes the total error (E) generated by a network to a set of training patterns (e.g., Dawson, 2004). In particular, the error that is minimized is the sum of squared differences between desired and observed activities, taken over the total number of output units, and over the total number of training patterns. In the equation below, there are *m* total patterns, *n* total output units, t_{pj} is the desired activity of output unit *j* to pattern *p*, and a_{pj} is the actual activity of this unit to this pattern.

$$E = \sum_{p=1,m} \sum_{j=1,n} (t_{pj} - a_{pj})^2 \quad (39)$$

Dawson and Schopflocher (1992) found that to train a value unit an elaborated error term had to be minimized. This term included the equation above, but added to it an additional error component that measured the difference between the net input of an output unit to a pattern (net_{pj}) and the current mean of the unit's activation function (μ_j) for patterns that were supposed to activate the output unit. The elaborated error term is:

$$E = \sum_{p=1,m} \sum_{j=1,n} (t_{pj} - a_{pj})^2 + \sum_{p=1,m} \sum_{j=1,n} t_{pj} (net_{pj} - \mu_{pj})^2 \quad (40)$$

6.12.2 Gradient Descent

Armed with their definition of error, Dawson and Schopflocher were able to derive a gradient descent rule that could be used to train a value unit. Because this was a gradient descent rule, it included the derivative of the Gaussian, G'(net_{pj}):

$$G'(net_{pj}) = -2\pi \cdot (net_{pj}) \cdot G(net_{pj}) = -2\pi \cdot (net_{pj}) \cdot e^{(-\pi (net_{pj} - \mu)^2)} \quad (41)$$

They defined a learning rule error term for the first part of their error equation as δ_{pj}, and discovered that it was identical to the gradient descent term for the integration device, with the exception that a different derivative was used:

$$\delta_{pj} = (t_{pj} - a_{pj}) \cdot G'(net_{pj}) \quad (42)$$

Dawson and Schopflocher (1992) also defined an error term (ε_{pj})that was related to the second component of their error equation:

$$\varepsilon_{pj} = t_{pj} \cdot (net_{pj} - \mu_{pj}) \quad (43)$$

The complete learning rule for modifying a weight using both of these error terms could then be expressed in a format that makes apparent its close relationship to the delta rule and to the gradient descent rule for an integration device. In the equation below, η is the learning rate:

$$\Delta w_{ij} = \eta \cdot (\delta_{pj} - \varepsilon_{pj}) \cdot a_{pj} \quad (44)$$

As was the case for the integration device, this equation can be used to modify μ_j by treating it as the weight from an input unit that is always on.

6.12.3 An Example Network

A value unit network can be trained, using this learning rule, to solve XOR without the need for hidden units. Table 6-10 presents the structure of an example network that solved this problem after 50 sweeps with a learning rate of 0.10. Note that this network solves the problem because the only time that the net input coming from the input units is equal to μ is when only one of the input units is on. If both are on, or if both are off, the net input is sufficiently far from μ to turn the output unit off.

Unit	Weight	μ
X	-0.87	-
Y	-0.87	-
Output	-	-0.86

Table 6-10

6.13 Implications for Animal Learning

6.13.1 Design Decisions

This chapter has examined three different design decisions that are part of using artificial neural networks to simulate aspects of animal learning. First, it argued that the bias term in a perceptron provided a means to model associations to uncontrolled background stimuli. Second, it argued provided empirical and logical arguments that a null condition, in which the perceptron was trained to turn off when no input units were activated, should be included in these sorts of simulations. Third, it demonstrated that one could easily change the capabilities of a perceptron – and deal with linearly nonseparable problems such as negative patterning – by replacing a monotonic activation function (e.g., the logistic equation) with a non-monotonic activation function (e.g., the Gaussian equation).

With respect to this latter design decision, it is important to recognize that the learning rules that have been proposed for the three architectural variants (perceptron, integration device, and value unit) are all structurally identical. First, all three learning rules are error correcting. Second, all three learning rules define a change in weight in terms of the product of three different values: a learning rate, a measure of output unit error, and an input unit's activity. The only difference between the three rules is that output unit error is defined with slightly different equations.

The structural similarity of the three rules carries over to the proven relationship between each of them and the Rescorla-Wager model. In Section 5.8, we proved that one could translate delta rule learning into the Rescorla-Wagner equation. Furthermore, we proved this when it was assumed that the activation function for the device being trained was nonlinear. Finally, this proof of the equivalence between the two approaches did not depend upon the precise nature of the nonlinear activation function. What this means is that any of the three versions of the perceptron that we have explored thus far in this book have a formal equivalence to the Rescorla-Wagner model. The source of this equivalence is the fact that they are trained with a variant of the delta rule.

6.13.2 A Problem Emerges

This established formal equivalence between our three architectures and the Rescorla-Wagner model raises a surprising problem: if all of these systems are equivalent, then how is it possible for some of them to be able to solve a different set of problems than the others? That is, if integration devices, value units, and Rescorla-Wagner learning are functionally equivalent, then how is it possible for two of these systems to be incapable of solving negative patterning, while the third can? It would appear that systems that are formally equivalent at one level can respond differently at another.

It could, of course, be argued that this problem only arises because of the special properties of the value unit. However, we will see in the next chapter that this argument is false. This is because we will see that there are several learning paradigms for which the Rescorla-Wagner model makes different predictions than does the integration device. In other words, the problem doesn't arise because of the specific properties of one artificial neural network. Rather, the assumed equivalence between these networks and Rescorla-Wagner is not completely accurate. We now turn to examining this problem both empirically and theoretically.

Chapter 7: The Paradox of the Perceptron

Previous chapters have provided empirical and formal results that have demonstrated the equivalence of the delta rule and the Rescorla-Wagner model. However, without a great deal of effort one can raise serious questions about this relationship. In particular, it is fairly easy to generate results in which an integration device responds in a fashion that is contrary to the predictions of the Rescorla-Wagner model. The purpose of this chapter is to provide the reader with several such results, and to raise deeper questions about the relationship between the models of network and animal learning that have been discussed to this point.

A peer-reviewed monograph published by Comparative Cognition and Behavior Reviews on behalf of the Comparative Cognition Society.

7.1 Old Connectionism and the New Associationism

7.1.1 Two Connectionist Eras

It is common to divide connectionism into two historical eras, Old Connectionism and New Connectionism (Medler, 1998). New Connectionism concerns the study of adaptive, multilayered networks, and arose in the mid-1980s (Rumelhart & McClelland, 1986).

The networks that we have been studying in this book all properly belong to the earlier era of Old Connectionism, which lasted until the late 1960s. The adaptive networks of Old Connectionism were not multilayered – they only included input units and output units, and had no intermediate processors. This is because researchers were unable to determine how multilayer networks could be trained. Bernard Widrow, one of the leading figures of Old Connectionism, recalled "we stopped doing neural nets because we'd hit a brick wall trying to adapt multilayer nets" (Anderson & Rosenfeld, 1998).

We have seen that the limitations of the standard perceptron and of the integration device rest with the fact that such networks can only solve linearly separable problems (Minsky & Papert, 1988). This problem is directly related to the absence of intermediate processors. However, it is easy to establish that animals and humans can learn to solve linearly separable problems (Shanks, 1995).

For this reason, most modern studies of artificial neural networks examine the properties of systems that have one or more layers of hidden units. Such networks are far more powerful than the perceptron, and became extremely popular after researchers discovered learning rules that could be used to train them (Rumelhart et al., 1986). "For example, we know that connectionist networks have the in principle power of a universal Turing machine, but we also know that perceptron-like single layer networks are limited in their computational power. Thus, we should focus current research on multilayer networks" (Medler, 1998, p. 40).

7.1.2 New Associationism

The limitations of Old Connectionist networks have not been lost on researchers who are interested in associative learning (Quinlan, 1991; Shanks, 1995). This is particularly true given the formal links that we have seen between the delta rule and the Rescorla-Wagner model. For the most part, modern attempts to relate artificial neural networks to associative learning do so using multilayer networks (Enquist & Ghirlanda, 2005; Schmajuk, 1997). One might view such research as representing a modern revitalization of associationism (Bechtel, 1985; Bechtel & Abrahamsen, 1991).

It is almost certainly the case that many of the regularities in associative learning will only be captured by employing multilayer networks trained with modern learning rules. However, it is also the case that the dismissal of simpler networks and their relationship to associative learning is premature. One reason for this is that many of these phenomena may only require the kind of explanation that can be couched in terms of Old Connectionism. A second reason is that we have already seen that modern variations of the perceptron, such as the value unit, are capable of solving some linearly nonseparable problems.

A third reason for continuing to explore the relationship between Old Connectionism and associative learning is that it is far more complicated than one would imagine. In particular, in spite of the empirical and formal results that have been presented thus far in this book, it is relatively easy to demonstrate that Old Connectionist networks can generate results that are contrary to the predictions of the Rescorla-Wagner model. The fact that an integration device can generate such discrepancies, in light of the formal results of Chapter 5, provides the paradox of the perceptron. The purpose of this chapter is to provide several demonstrations of this paradox. An account of the paradox, and of its implications for theories of associative learning, follows in Chapters 8 and 9.

7.2 Facilitated Reacquisition

7.2.1 The Savings Phenomenon

Consider a paradigm in which an association is learned, then extinguished, and finally learned again. A common finding in the literature is that the reestablishment of the association occurs faster than does the original conditioning (Napier, Macrae, & Kehoe, 1992). This effect is called facilitated reacquisition. This is a form of savings, in the sense that there must be some of the original learning saved after extinction in order to facilitate relearning.

Miller, Barnet, and Grahame (1995) have pointed out that one of the failures of the Rescorla-Wagner model is that it cannot account for facilitated reacquisition. This is because the Rescorla-Wagner model assumes that extinction is the process by which an associative strength becomes 0.

7.2.2 Method and Results

A simulation was conducted to determine whether facilitated reacquisition could be observed in an integration device. 25 different networks were run as subjects in a study in which independent associations involving CS_A and CS_B were established in Phase 1, extinguished in Phase 2, and reestablished in Phase 3. The simulation settings for this study, as well as the remaining ones described in this chapter, were identical to those used in the Chapter 4 simulations. The training sets for the current study are provided in Table 7-1:

Phase	Trial Type	CS_A	CS_B	t_j
1	-	0	0	**0**
	A+	1	0	**1**
	B+	0	1	**1**
2	-	0	0	**0**
	A-	1	0	**0**
	B-	0	1	**0**
3	-	0	0	**0**
	A+	1	0	**1**
	B+	0	1	**1**

Table 7-1

The results of this experiment demonstrated strong facilitated reacquisition. On average, networks learned the associations to the CSs after 695.9 sweeps during Phase 1. After extinction, networks reacquired these associations after only 62.6 sweeps. This difference was statistically significant ($t = 3005.415$, $df = 48$, $p < 0.0001$).

Table 7-2 provides the structure of a typical network after the three different phases of learning. An examination of this table explains why facilitated reacquisition is observed in an integration device. The effect of the first phase of training is to turn each CS into a strong conditioned excitor, and background context into a strong conditioned inhibitor (to turn the integration device when it is subjected to the null condition). Interestingly, the effect of the second phase of training is to slightly reduce the associations for the two CSs, and to convert background context into an extremely powerful conditioned inhibitor – powerful enough to overwhelm the excitatory signal coming from either CS. Because the weights for the two CSs remain highly positive, much less training is required to make the integration device respond in the desired fashion during the third phase of training.

Phase	Unit	Weight	θ
1	CS_A	4.92	-
	CS_B	4.92	-
	Output	-	-2.20
2	CS_A	3.28	-
	CS_B	3.29	-
	Output	-	-5.50
3	CS_A	4.78	-
	CS_B	4.78	-
	Output	-	-2.57

Table 7-2

The key result of this simulation is that facilitated reacquisition is easily demonstrated in this particular model of network learning.

7.3 Retarded Reacquisition

7.3.1 An Alternative Effect

Extinguishing conditioning to a CS can lead to a situation in which reacquisition is facilitated. However, in some instances the reverse is true.

In one example study, rats underwent fear conditioning by having a tone (CS) paired with a shock (US) (Bouton, 1986). This conditioning was then extinguished. When animals were subjected to a moderate amount of extinction (16 or 24 trials), reacquisition of the fear conditioning proceeded at the same rate as control animals. However, when animals experienced a prolonged period of extinction (72 trials), reacquisition of the fear conditioning proceeded significantly slower than was the case for control animals. The Rescorla-Wagner model does not predict this result for exactly the same reasons that the model does not predict facilitated reacquisition (Miller, Barnet & Grahame, 1995, p. 371).

7.3.2 Method and Results

Interestingly, a minor change in the methodology that was used in the previous simulation produces retarded reacquisition. As was noted in Section 7.2, a great deal of the facilitated reacquisition effect was due to changes in network bias. It was previously shown that a network's bias is strongly affected by the null condition (compare Section 4.10.2 to Section 6.6.2). The current simulation removed the null condition (Table 7-3), and then repeated the design described in Section 7.2.

Phase	Trial Type	CS_A	CS_B	t_j
1	A+	1	0	1
	B+	0	1	1
2	A-	1	0	0
	B-	0	1	0
3	A+	1	0	1
	B+	0	1	1

Table 7-3

The exclusion of the null condition in this simulation produced a much different effect than the one observed in Section 7.2. On average, networks converged on Phase 1 training after 48.12 sweeps. After extinction, Phase 3 reacquisition of the conditioning to the two CSs was longer, requiring an average of 59.72 sweeps. This difference was statistically significant ($t = 71.756$, $df = 48$, $p < 0.0001$).

7.3.3 Explanation and Implications

Why is reacquisition retarded in this simulation? Table 7-4 provides the structure of a typical network after each training phase. After Phase 1, both CSs and background context have all become conditioned excitors. This differs from the previous simulation, in which background context became a conditioned inhibitor. After Phase 2, all three have become conditioned inhibitors. Again, this differs from the preceding simulation, in which CS_A and CS_B remained as conditioned excitors. Phase 3 training converts all three back into conditioned excitors. Because all three network components must be changed from having negative to positive values after Phase 2, learning takes longer in Phase 3 than does the initial Phase 2 learning.

Phase	Unit	Weight	θ
1	CS_A	0.77	-
	CS_B	0.77	-
	Output	-	1.43
2	CS_A	-0.69	-
	CS_B	-0.70	-
	Output	-	-1.51
3	CS_A	0.77	-
	CS_B	0.77	-
	Output	-	1.43

Table 7-4

The simulation methodology is quite different from that used by Bouton (1986). As a result, different explanations of the effect are provided by the two different approaches. Bouton makes an argument that CSs retain excitatory associations, but when these are very weak, retarded reacquisition results. The simulation explains retarded reacquisition by noting that the effect of extinction is to convert the CSs into conditioned inhibitors.

7.4 Extinction of a Conditioned Inhibitor

7.4.1 Undesired Extinction

In the Rescorla-Wagner model there is symmetry between excitation and inhibition. This is because "conditioned excitation and inhibition reflect positive and negative values, respectively, of a common variable representing associative strength" (Miller, Barnet, & Grahame, 1995, p. 371). In Section 4.3, it was shown that extinction of a conditioned excitor in an integration device causes its associative strength to move in a negative direction, which is consistent with the Rescorla-Wagner model. The symmetry of the Rescorla-Wagner model predicts, then, that extinction of a conditioned inhibitor should cause its associative strength to move in a positive direction.

Unfortunately, this prediction is not borne out in the animal literature (e.g., Zimmer-Hart & Rescorla, 1974). Miller, Barnet and Grahame (1995) cite several studies that indicate that if a conditioned inhibitor is presented alone (i.e., without the US), its associative strength is either unchanged, or moves in a negative direction. "These observations are among the more problematic for the Rescorla-Wagner model" (p. 371).

7.4.2 Method and Results

A simulation study was conducted to determine how a conditioned inhibitor responded to extinction in an integration device. Two phases of training were conducted to first make CS_A a conditioned excitor and CS_B a conditioned inhibitor, and to then extinguish this training (see Table 7-5). The question of interest is the effect of this extinction on the weight from CS_B.

Phase	Trial Type	CS_A	CS_B	t_j
1	-	0	0	0
	A+	1	0	1
	AB-	1	1	0
2	-	0	0	0
	A-	1	0	0
	B-	0	1	0

Table 7-5

Table 7-6 provides the structure of a network averaged over 25 different integration devices that were used as subjects in this study. It can be seen that the first phase of training resulted a strong excitatory weight for CS_A, a strong inhibitory weight for CS_B, and a moderately negative bias. The extinction phase of training decreased the weight for CS_A ($t = 23.963$, $df = 48$, $p < 0.0001$), made the bias much more negative ($t = 23.963$, $df = 48$, $p < 0.0001$), made the weight for CS_B marginally more negative. This last effect was very slight in absolute terms, but was statistically significant ($t = 18.141$, $df = 48$, $p < 0.0001$) because this weight always was made more negative after the second phase of training.

Phase	Unit	Weight	θ
1	CS_A	4.47	-
	CS_B	-4.80	-
	Output	-	-2.45
2	CS_A	2.37	-
	CS_B	-4.81	-
	Output	-	-4.58

Table 7-6

These results are consistent with the Rescorla-Wagner model as far as the conditioned excitor is concerned: it becomes less excitatory after extinction. However, they are inconsistent with this model, and are consistent with the animal learning results, as far as the conditioned inhibitor is concerned: the weight for CS_B does not move in a positive direction as the Rescorla-Wagner model predicts.

Unlike the simulations discussed in Sections 7.2 and 7.3, this result is not dependent upon the presence of the null condition. If the simulation is carried out with the null condition removed from both training sets, it is still the case that the second phase of training does not change the weight for CS_B in a positive direction.

7.5 Conditioned Inhibition and Nonreinforced Novel Cues

7.5.1 Conditioned Inhibition

Pavlov (1927) demonstrated that conditioned inhibition can be produced by reinforcing CS_A, but not reinforcing the compound stimulus CS_{AB}. "In this way the combination is gradually rendered ineffective, so that the conditioned stimulus when applied in combination with the additional stimulus loses its positive effect, although when applied singly and with constant reinforcement it retains its full powers (Pavlov, 1927, p. 68). The agent is inhibited by CS_B, because it has learned that it is a signal that reinforcement is not forthcoming.

This paradigm can easily be used to produce conditioned inhibition in an integration device that employs the logistic activation function. At the end of such training (requiring in the order of 550 sweeps), a typical network would have a bias of -2.36, a weight of 4.56 for CS_A, and a weight of -4.80 for CS_B.

7.5.2 Adding a Novel Cue

After using the procedure above to turn CS_B into a conditioned inhibitor, one could then proceed to introduce a third, novel stimulus (CS_C). This novel stimulus could be repeatedly paired with the conditioned inhibitor CS_B, with the pair never being reinforced. This training would proceed without reinitializing the weights, so that the previous effects of learning were preserved in the network when the novel stimulus was introduced.

Miller, Barnet and Grahame (1995) have noted that the Rescorla-Wagner model predicts that this pairing should convert CS_C into a conditioned excitor. This is because of the model's presumed symmetry between excitation and inhibition, which was discussed in Section 7.4. In other words, this second phase of learning is simply a mirror image of the kind of learning that was used to make CS_B a conditioned inhibitor, and therefore should produce the opposite result.

However, this prediction counts as a failure of the model. Baker (1974), for instance, conducted a carefully controlled experiment that failed to provide any evidence that the second phase of training caused the novel stimulus to become excitatory.

In contrast to the prediction of the Rescorla-Wagner model, the second phase of network training involving CS_B and CS_C does not turn the latter into an excitor. This is because in this training, when these two stimuli are presented together, the desired response is to have the output unit turn off. Importantly, this desired response is produced almost immediately because of the strong inhibitory weight that has already been produced to CS_B. As a result, the network converges quickly, and has had little opportunity to modify the weight for CS_C, which typically starts as a small random number according to the methods that have been used throughout this manuscript. For example, the integration device described in Section 4.5.1 converged after 1 sweep of additional training, and produced a final weight of 0.05 for CS_C.

As was the case in Section 7.4, the reason that the network makes a different prediction than does the Rescorla-Wagner rule is because network learning is driven by the output unit's response, and not by underlying associative strengths. Output unit responses do not preserve the symmetry between excitation and inhibition that are central to the dynamics of the Rescorla-Wagner rule, and this breaking of symmetry permits the network to predict results that are more consistent with existing results from the experimental literature.

7.6 Latent Inhibition

7.6.1 The CS-Preexposure Effect

In latent inhibition (Lubow & Moore, 1959), or the CS-preexposure effect, a CS is presented in the absence of the US. This preexposure produces slower conditioning to the CS than in a control condition in which the preexposure does not occur. The standard account of latent inhibition is in terms of modulations of attentional resources devoted to the CS (Lubow, 2005).

Latent inhibition is a phenomenon that is not predicted by the Rescorla-Wagner model (Miller, Barnet, & Grahame, 1995, p. 373). This is because the initial association for the CS is 0, and therefore should not change when the λ for nonreinforced trials is also equal to 0.

7.6.2 Method

Can latent inhibition be produced in an integration device? Two training sets were developed to answer this question, and are presented in Table 7-7. The first included the null condition and the presentation of CS_A without reinforcement. The second included the null condition and the presentation of CS_A with reinforcement. Control networks were only trained with the second set of patterns; experimental networks were trained on the first, and then received continued training on the second without altering the weights produced by the first phase of training.

Phase	Trial Type	CS_A	t_j
Preexposure	-	0	0
	A-	1	0
Learning	-	0	0
	A+	1	1

Table 7-7

7.6.3 Results

On average, 25 different networks in the control condition learned to respond to CS_A after 421.0 sweeps. When experimental networks were preexposed to CS_A without reinforcement, they learned to respond to CSA in Phase 2 after 428.4 sweeps. While this difference was small, it was extremely reliable: no experimental network learned the association as quickly as the slowest control network. As a result, this effect was statistically significant ($t = 23.115$, $df = 48$, $p < 0.0001$). In other words, while the Rescorla-Wagner model does not produce latent inhibition, the integration device does.

7.6.4 Explaining the Effect

How is latent inhibition produced in the integration device? Table 7-8 presents the structure of a typical network produced from preexposure to CS_A, as well as the structure of a typical network that has learned to respond to CS_A. It can be seen that the main effect of preexposure is to turn the context into a strong conditioned inhibitor (as evidenced by the negative bias), and to turn CS_A into a milder conditioned inhibitor (as evidence by its less extreme negative weight). In order to learn to respond to CS_A, the weight from its input unit must be much larger and positive, as is also seen in the table. The fact that CS_A is turned into a mild conditioned inhibitor by Phase 2 learning means that more learning is required to convert this negative weight into a stronger one. Without preexposure, the weight from CS_A would be randomly started in the range from -0.1 to 0.1, and therefore less learning would be required to turn the stimulus into a conditioned excitor.

Type of Network	Weight from CS_A	θ
Preexposure	-0.77	-2.20
Conditioned to CS_A	4.63	-2.20

Table 7-8

There are many theories about latent inhibition, as well as a complex set of empirical findings that any single theory has difficulty capturing completely (e.g., Schmajuk, Lam, & Gray, 1996). The network results contribute to this mix by suggesting that in some instances latent inhibition might be due to the fact that preexposure causes the CS to become a conditioned inhibitor.

7.7 Superblocking

7.7.1 A Strange Prediction

For the most part, we have been examining the performance of the perceptrons empirically, by examining the results of computer simulations. Some researchers have argued that if one can use simulations to make a point about a theory, then one should be able to make the same point theoretically (Kukla, 1989). This section attempts to do so by arguing that an integration device will not make a particularly counterintuitive prediction that is made by the Rescorla-Wagner model.

Consider a system that has received conditioning such that CS_A and CS_B have been presented independently along with the US. As a result, both of these CSs have become conditioned excitors. Now imagine that they are both presented at the same time along with a new stimulus, CS_C; let this compound of three CSs be reinforced. What will happen to the association for CS_C?

Miller, Barnet, and Grahame (1995) argued that the Rescorla-Wagner model predicts that this situation will produce superblocking of CS_C. That is, the model does not just predict that the previous training involving the other two CSs will block the conversion of CS_C into a conditioned excitor. Instead, the Rescorla-Wagner model actually predicts superconditioning: when the compound stimulus is presented, CS_C will become a conditioned inhibitor! They list this counterintuitive prediction as a failure of the Rescorla-Wagner model because there is scarce evidence in the animal literature for such a superblocking effect.

7.7.2 A Thought Experiment

Let us conduct a thought experiment to determine whether superblocking will occur in an integration device. The subject in this thought experiment will be a network that has one output unit and three input units. The three input units are used to represent the presence or absence of three different conditioned stimuli: CS_A, CS_B, and CS_C.

Imagine that in Phase 1 of training, this network conditioned as follows: -, A+, and B+. When this training is completed, the network will turn on when either CS_A or CS_B is presented alone, and will turn off to the null condition. This means that the bias will be negative enough to produce low activity (0.10) when no CSs are present. Furthermore, connection weights for CS_A and CS_B must both be sufficiently high to produce a net input that will overcome the bias, and produce output unit activation that is at least 0.90. Finally, because CS_C was not presented, the weight from CS_C will not have been trained, and thus will be whatever small random value that the weight was initialized to before Phase 1 training began.

This network now undergoes Phase 2 training, during which it is given different conditioning: - and ABC+. What will happen? Previous training has already caused the network to turn off to the null condition, so when the null condition is presented in Phase 2 it will still turn off – no additional training is required to generate a hit. Furthermore, previous training has produced a network that will turn on to CS_A or CS_B alone. So, when they are presented together in Phase 2 (the presence of CS_C will be irrelevant), the net input will be even higher than was the case in Phase 1. As a result, the network will turn on to the compound stimulus without any additional training.

In short, the network will generate hits to both Phase 2 patterns without any additional training. Because of this, the learning rule *will not alter the connection weights*. They will remain identical to the weights that were produced in Phase 1.

This clearly can be interpreted as a situation in which prior learning on CS_A and CS_B blocks later learning involving CS_C. However, it also shows that superblocking will not occur in the network. The fact that Phase 2 will not cause any further changes in connection weights indicates that CS_C will not become a conditioned inhibitor. The thought experiment has revealed another discrepancy between the integration device and the Rescorla-Wagner model.

7.8 The Overexpectation Effect

7.8.1 Overexpectation

To this point in the chapter, we have examined five different examples in which the responses of an integration device diverge from the predictions of the Rescorla-Wagner model. For four of these results, the responses of the integration device might be seen as an improvement, because network performance corrects what Miller, Barnet, and Grahame (1995) list as a failure of the Rescorla-Wagner model. This is not the intended interpretation. To emphasize this point, let us turn to a discrepancy in which integration devices fail to produce one of the many successes of the animal learning theory.

One of the most theoretically important effects in associative learning is the overexpectation effect. The effect is produced when CS_A and CS_B have been independently paired with a US. Then, the two CSs are presented as a compound and are paired with the US. Overexpectation is defined as occurring when there is reduced responding (relative to a control) to CS_A and CS_B as individual stimuli following the training on the compound stimulus.

This result is counterintuitive because CS_A and CS_B have apparently lost associative strength despite continued reinforcement during the compound training. Nevertheless, overexpectation effects have been found in studies on Pavlovian fear conditioning in rats (Blaisdell, Denniston, & Miller, 2001; Kremer, 1978; Rescorla, 1970), appetitive conditioning in rats (Lattal & Nakajima, 1998; Rescorla, 1999) and autoshaping with pigeons (Khallad & Moore, 1996). Furthermore, overexpectation is predicted from the Rescorla-Wagner model, and counts as one of its most notable successes (Miller, Barnet & Grahame, 1995).

The Rescorla-Wagner model predicts overexpectation because the result of conditioning to the two individual CSs is that the associative strength for each will be near λ. Therefore, at the start of training to the compound stimulus, the ΣV term in the model will be near 2λ, because each CS will be contributing λ to this term. One consequence of this is that the expression $(\lambda - \Sigma V)$ in the Rescorla-Wagner model will take on a negative value. As a result, the associative strengths for each individual CS will be decreased when training to the compound begins. Later, when the CSs are presented individually, weaker responses are expected because of these decreased associative strengths.

7.8.2 A Network Failure

An argument identical in structure to the one presented in Section 7.6.2 concludes that an integration device will not produce the overexpectation effect (Dawson & Spetch, 2005). Dawson and Spetch imagined a network that, in Phase 1 of training, learned to respond to CS_A or CS_B (the compound CS_{AB} was not presented), and learned not to respond in their absence. Such a network is very similar to the one described in Section 7.6 after Phase 1 training. Not surprisingly, the individual connection weights from both CS_A and CS_B to the output unit must each be substantially larger than the bias. This is because the presence of only one of these CSs must produce a signal that is sufficiently strong to turn the output unit on.

Dawson and Spetch (2005) proposed that in Phase 2 of training the network is presented CS_A and CS_B together as a compound stimulus. As was argued in Section 7.6.2, this will result in an even stronger signal being sent to the output unit, which will cause it to (correctly) turn on. Because this response is correct – and because the network will correctly fail to respond to the null condition – the network's weights and bias will not be modified at all during Phase 2. Therefore the perceptron's response to the individual stimuli will not decrease, and the overexpectation effect will not be produced.

Dawson and Spetch (2005) reported a simulation that supported this prediction. Thus overexpectation is a discrepancy between machine learning and animal learning that counts as a failure of the integration device.

7.9 Overexpectation in a Value Unit

7.9.1 Changing the Network

An integration device will not generate the overexpectation effect because its activation function is monotonic. Once the connection weights have been adjusted to produce a net input that will turn the output unit on, any change that produces an increase in net input will still turn the output unit on. If this is the correct response, then weights will not be changed.

In Chapter 6, it was noted that choosing the activation function is a design decision faced by a neural network researcher. One could replace the logistic activation function with the Gaussian. Because the Gaussian is nonmonotonic, it is possible that a value unit may exhibit the overexpectation effect.

7.9.2 A Network Success

Dawson and Spetch (2005) confirmed this possibility. They found that when the compound stimulus was presented to a value unit that had already learned to respond to CS_A and CS_B presented separately, the value unit incorrectly turned off. This error caused the learning rule to modify the connection weights. At the end of a small amount of training, the value unit responded correctly to the compound stimulus (and also responded correctly to the null condition).

Importantly, this small amount of additional training also produced the overexpectation effect. A value unit generated activities of less than 0.50 to either CS_A or CS_B when they were presented independently. Recall that this was after Phase 1 training had caused the network to turn on to either of these stimuli.

7.9.3 Implications of this Result

Dawson and Spetch's (2005) demonstration that a value unit will produce the overexpectation effect should not be taken as suggesting that this type of network provides a better model of conditioning because of a better fit to empirical predictions of the Rescorla-Wagner model. It merely shows that altering a design decision can produce radically different responses in a perceptron. One version of the perceptron produces responses that are consistent with a model of animal learning, while a different version of the same type of network produces results that are not.

This sort of result could motivate a research program in which one systematically modified the properties of a neural network in an attempt to find the architecture that provided an optimal fit to animal data, or the highest degree of concordance with an animal learning model, or both. Such a research program would almost certainly produce an interesting set of results.

For instance, it would likely uncover a set of interactions between network types and problem types (e.g., Dawson, 2005, Chap. 14). That is, one version of the perceptron will handle some problems nicely, but will have difficulties with others. Changing the perceptron might help with these difficulties, but reveal a different set of troubles for the new network. For instance, the integration device has problems with overexpectation. Changing the integration device to a value unit solves this problem, but will raise others. For instance, a value unit might mistakenly predict superblocking.

Network type by problem type interactions might lead researchers to consider principles methods for deciding what type of perceptron is most appropriate for modeling associative learning. However, researchers would benefit more from first considering a more general question. Given the formal equivalence between these networks and the Rescorla-Wagner model, how is it possible for perceptrons to generate results that differ from Rescorla-Wagner predictions? This is the paradox of the perceptron – how does formal equivalence lead to empirical discrepancy?

In addition to the formal relationships between these two approaches to learning, there must also be fundamental differences between them. What are these differences, and how might they impact theories of learning? The purpose of Chapter 8 is to examine such questions.

Chapter 8: Models, Simulations, and Behavior

The previous chapter demonstrated that in spite of the formal equivalence that was established in Chapter 5, a number of cases exist where a particular perceptron – the integration device – generates different results than does the Rescorla-Wagner model. Why does this perceptron paradox exist? The current chapter argues that the perceptron paradox arises because the Rescorla-Wagner model and the perceptron are qualitatively different types of models. It explores the notion of different types of models in psychology and cognitive science, and how these differences lead to the paradox. The paradox is resolved in the realization that the perceptron, as a computer simulation, must respond before its connection weights are updated. In contrast, the Rescorla-Wagner equation defines how associations can be altered without first requiring responses. It is argued that while the two might be equivalent at the computational level of analysis, they are not at the algorithmic level, where the effects of responding (or not) are critical.

A peer-reviewed monograph published by Comparative Cognition and Behavior Reviews on behalf of the Comparative Cognition Society.

8.1 A Multitude of Models

Modern psychology has made modeling one of its major methodologies. However, it has been argued that not all of the models that it uses are of the same type (Dawson, 2004). This argument plays a central role in understanding the perceptron paradox that was detailed in Chapter 7. For this reason, let us briefly consider the major classes of psychological models.

8.1.1 Models for Data

The type of model that is most used in psychology is a model for data. “Models for data summarize a set of observations in the behavioral or biological sciences so that we may communicate with our colleagues and the public” (Lunneborg, 1994). In other words, all of the statistical tools that are employed by experimental psychologists are models of data. As such, they fit preexisting observations or measurements, they are usually linear, they are ordinarily evaluated by considering their goodness of fit to the existing observations, they can describe behavior but do not behave in their own right, and they are not designed to surprise us (Dawson, 2004). Neither the Rescorla-Wagner model nor the perceptron are models for data, and therefore this type of model will not concern us for the rest of this chapter.

8.1.2 Mathematical Models

The Rescorla-Wagner model is a prototypical example of a mathematical model in psychology. "From the first efforts toward psychological measurement, investigators have had in mind the goal of making progress toward generality in psychological theory by developing quantities analogous to mass, charge, and the like in physics and showing that laws and principals formulated in terms of these derive quantities would have greater generality than those formulated in terms of observables" (Estes, 1975). In other words, mathematical models have a much more ambitious goal than do models of data: mathematical models are intended to provide a quantitative expression of psychological laws that not only describe the primary data from which they were derived, but can also be used to generate new experiments that can often confirm a mathematical model’s surprising prediction.

One reason for the lasting influence of the Rescorla-Wagner model was its ability to generate a number of surprising predictions, many of which were confirmed by new experiments that were themselves inspired by the model (Miller et al., 1995). For example, the model was originally derived to provide an account of the blocking phenomenon (Kamin, 1968, 1969). However, its validity was established when it predicted a number of counterintuitive situations to “unblock” stimuli, and had these predictions experimentally confirmed (Walkenbach & Haddad, 1980).

8.1.3 Computer Simulations

The various perceptrons that have been explored in this book are examples of computer simulations. At the most basic level, a computer simulation in psychology is an effective procedure for actually generating behavior of interest. “Since cognitive scientists aim to understand the human mind, they, too, must construct a working model” (Johnson-Laird, 1983). We will see that a crucial characteristic of this type of model is that it doesn’t just describe a response, but it actually produces it (Dawson, 2004).

Computer simulations share many of the characteristics of mathematical models, but have been sited as causing the decline of mathematical psychology. "Learning to live with computers is perhaps the single most difficult and critical task facing mathematical psychology as a discipline" (Estes, 1975, p. 267). This is because computers have "made it relatively easy to simulate quite complex interactive systems. For many, it is clearly simpler and more agreeable to program then it is to study processes mathematically" (Luce, 1997). The tension between mathematical models and computer simulations is at the heart of the perceptron paradox, and must be considered in more detail in this chapter.

8.2 Rescorla-Wagner as a Mathematical Model

Mathematical models, like other models used in psychology, can be characterized in terms of five different major features (Dawson, 2004). As the Rescorla-Wagner model is a prototypical mathematical model, we can consider it in terms of these five key characteristics.

8.2.1 Nonlinearity

Iteration is a mathematical technique that can be described as repetition with a purpose (Kovach, 1964). A process is iterative when the result that it produces at time *t* is used as an input for processing at time *t+1*. The Rescorla-Wagner model is iterative in exactly this sense. At time t, the model (e.g., Equation 18 in Section 5.4) is used to calculate a change in associative strength (e.g., $\Delta V_{a(t)}$). This change is then applied, resulting in a new value for overall associative strength (ΣV). This new value for ΣV is then used in the next iteration of the equation to calculate a new output value (e.g., $\Delta V_{a(t+1)}$).

The equation for the Rescorla-Wagner model appears to be linear, because it is only composed of simple multiplication and subtraction. However, when this equation is used iteratively, its results are highly nonlinear. For instance, the model predicts that acquisition curves will not be straight lines, but will instead be exponentially decreasing curves.

8.2.2 Surprise

"What we hope for primarily from models is that they will bring out relationships between experiments or sets of data that we would not otherwise have perceived" (Estes, 1975, p. 271). Linear models, such as models of data, are rarely capable of doing this. To say that a model is linear is to say that one can predict the behavior of the entire system from knowing its components (Luce, 1999). Surprises emerge when linearity is replaced by nonlinearity, because then knowledge of the individual components is not sufficient for understanding the whole system (Dawson, 2004). The intrinsic nonlinearity of the Rescorla-Wagner model is crucial for its ability to generate surprising – and accurate -- predictions.

8.2.3 Goodness of Fit

Mathematical models need to be evaluated by considering their ability to fit empirical observations. "Does not one simply construct a model, apply it to data, and accept or reject on the basis of goodness of fit? Well, that is indeed a standard procedure – perhaps the standard procedure" (Estes, 1975, p. 267). Of course, much of the success of the Rescorla-Wagner model rests with its ability to explain experimental results (Miller et al., 1995; Walkenbach & Haddad, 1980). Furthermore, to the extent that the model is challenged, the challenges begin by citing experimental results that the Rescorla-Wagner model does not fit well (Schmajuk & Larrauri, 2006).

8.2.4 Existing Data Required

In addition to being evaluated by examining a fit to observations, mathematical models usually originate from existing data. The Rescorla-Wagner model is no exception. It was inspired by considering existing experimental results – "the accumulation of a salient pattern of data, separate portions of which may be adequately handled by separate existing theories, but which appears to invite a more integrated theoretical account" (Rescorla & Wagner, 1972, p. 64).

8.2.5 Behavior

Dawson (2004) has argued that mathematical models describe behavior, but do not behave themselves. This is certainly true of modern theories of associative learning such as the Rescorla-Wagner model. "Consistent with the misguided name learning theory and inconsistent with the actual goal of explaining acquired behavior, most modern associative theories in the animal tradition emphasize the learning process (i.e., acquisition) per se and are virtually silent concerning the transformation of acquired information into behavior" (Miller, 2006, p. 82).

8.3 The Perceptron: Simulation, Not Mathematical Model

The various perceptrons that we have encountered are computer simulations, and not mathematical models (Dawson, 2004). Let us consider perceptrons in the context of the five features that were introduced in Section 8.2.

8.3.1 Nonlinearity

Perceptrons are nonlinear because of the activation function in the output unit – whether it is the Heaviside equation (perceptron), logistic equation (integration device), or Gaussian equation (value unit). The nonlinear activation function in a perceptron permits it to mediate stimulus-response relationships that cannot be modeled by simpler networks (e.g., Dawson, 2004), such as distributed associative memories.

8.3.2 Surprise

The nonlinear relationship between stimulus and response in a perceptron is clearly a source of surprise, both in terms of the responses that such networks can produce, and in terms of the internal structures that mediate these responses. "It's surprising how much computing can be done with a uniform network of simple interconnected elements" (Fodor & Pylyshyn, 1988). Hopefully the reader of the current manuscript has found surprises at least in the discrepancies between perceptrons and the Rescorla-Wagner model, and perhaps in the breadth of associative learning phenomena that these simple networks are capable of producing.

8.3.3 Goodness of Fit

Mathematical models are accepted or rejected on the basis of goodness of fit. Evaluating a computer simulation is more complicated. Many computer simulations are developed to solve problems, and not to fit data. "To me, the most troubling [trend] is some lack of concern about how complex computer models are to be evaluated empirically" (Luce, 1999, p. 733). Consider a perceptron trained in the overexpectation paradigm (Section 7.7). On the one hand, it does not produce the effect. On the other hand, it correctly makes all of the stimulus-response contingencies that it has been presented. Is this a poor or a good fit? Computer simulations "have no specifiable mathematical form and for which we are generally unable to formulate, let alone solve, the problem of testing goodness of fit" (Estes, 1975, p. 267).

8.3.4 Existing Data Required

In order to train a perceptron, one does not need to first conduct experiments to collect learning data (e.g., acquisition curves, response rates) to which the model will be fit. All that one needs is a set of stimulus-response pairings that will be used to guide the perceptron's responses during supervised learning. Dawson (2004) claims that artificial neural networks provide a medium in which one can conduct synthetic psychology, in which models are constructed prior to the collection of any empirical data.

8.3.5 Behavior

Dawson (2004) has argued that computer simulations are artifacts that not only describe behavior, but also produce it. This distinction is evident in early research on perceptrons. For example, "knobby ADALINE" was a physically realized perceptron that accepted inputs via electric switches that were set, and generated a physical response that categorized input patterns (Widrow, 1962). This machine didn't just learn – it learned to perform!

That this also distinguishes the perceptron from the Rescorla-Wagner model may be less evident, because both have been described as mathematical entities that change the values of different variables over time. However, in the perceptron output unit activity represents a response or a behavior (although we will see later that the meaning of this response or behavior is open to interpretation). In contrast, the Rescorla-Wagner equation does not contain *any* behavioral variables. Indeed, it defines how associations can be changed without requiring any response at all. This issue is discussed in detail in the next section.

8.4 What Did Rescorla and Wagner Model?

8.4.1 What Does The Model Describe?

What is the Rescorla-Wagner model a model of? In general, it has been described as a formal model of learning (Pearce, 1997), and as a model of Pavlovian conditioning (Walkenbach & Haddad, 1980). More specifically, the model "describes the learning curves for strength of association" (Rescorla & Wagner, 1972, p. 75).

The specifics of what the model describes can be ascertained by examining the equation that defines it (e.g., Equation 18, Section 5.4). At any given time, the equation defines the change in a particular value – the associative strength of a CS. When applied, this changes a particular CS's associative strength, as well as the value of overall associative strength. One can keep track of these different values, and plot them as a function of time. This is why Rescorla and Wagner describe their model as a model of learning curves.

In order to generate data of this type, one needs only specify the starting state of the system (i.e., a starting strength for each individual association) and the values for each of a small number of constants. Then the equation can be applied iteratively, computing the values of interest (i.e., associative strengths).

Clearly a particular instance of this process is designed to be meaningful. In particular, constants will be assigned to indicate whether stimuli are salient or not, and whether associations are being increased or extinguished. Furthermore, a researcher who uses the model is likely to attribute an interpretation to what each US and CS value represents (e.g., shocks, lights, etc.).

However, the assignment of such meaning is not required for the model to function. The model is a mathematical entity that manipulates a small set of variables according to the Rescorla-Wagner equation. The equation doesn't learn – it simply describes how certain variables change their values over time.

At the end of a particular use of the Rescorla-Wagner model, what can one say? The model will indicate the values of certain variables (e.g., the current strengths of a number of different associations). If these values are plotted over time, then the model provides information about their dynamics (e.g., Rescorla & Wagner, 1972, Figure 7). For example, it provides their direction and rate of change. Of course, this kind of quantitative information is crucial to evaluating the model.

8.4.2 What Doesn't It Describe?

Importantly, the Rescorla-Wagner model is not a model of behavior (Miller, 2006). It describes how associative strengths change over time. It does not describe anything else. At the end of using the model, one has the values of various associations. Presumably, these values – either alone or in combination – are important determinants of animal responses. However, the Rescorla-Wagner model does not provide any indication at all about how the numbers that it crunches are converted into responses. This is typical of mathematical models (Dawson, 2004).

Rescorla and Wagner (1972) are perfectly frank about the fact that theirs is not a model of response. "Independent assumptions will necessarily have to be made about the mapping of associative strengths into responding in any particular situation" (p. 75). Later (p. 77), they make this same point much more explicitly: "we need to provide some mapping of V values into behavior. We are not prepared to make detailed assumptions in this instance. In fact, we would assume that any such mapping would necessarily be peculiar to each experimental situation, and depend upon a large number of 'performance' variables."

We will see that a crucial difference between the Rescorla-Wagner model and the perceptron is that the former does not require responses to be elicited in order for associations to be modified.

8.5 Tacit Theories of Responding

8.5.1 Monotonic Theory of Response

Clearly Rescorla and Wagner (1972) wanted to compare their model to animal behavior. To do so, they needed to define the relationship between associative strength and response. They assumed that if one arranged a set of stimuli in order of their associative strength, and if one ordered the same stimuli in terms of the response magnitude that each stimulus produced, then the two orderings would be the same. They also assumed that if the associative strength of a stimulus was negative, then it would produce a response of zero magnitude.

This theory merely states that there is a monotonic relationship between association and response (Miller, 2006). Rescorla and Wagner assumed that increases in associative strength would not result in decreases in response magnitude. Such a relationship is not very precise, because it is consistent with several different functions that relate associative strength to behavior, such as either a linear or a sigmoid relationship between ΣV and response magnitude. Precise behavioral predictions would require a more definite theory (i.e., the specification of a particular output function).

Why is it, then, that such a theory does not accompany the Rescorla-Wagner model? It is because all of the learning in the model – all of the modifications of associative strength – occurs *in the absence of response*. That is, the Rescorla-Wagner equation specifies how associative changes can occur without the need for associations to be converted into observable responses. In order for Equation 18 of section 5.4 to work, one only needs information about associative strengths. At no time during learning do these strengths need to be converted into responses by specifying some additional output function.

8.5.2 Learning without Responding

An example of learning without responding is provided in the literature on probabilistic category learning (Gluck & Bower, 1988). In their modeling work, Gluck and Bower used an instance of the delta rule to train the weights of a perceptron-like network. (Their network was a perceptron with a linear activation function, because network output was equal to the output unit's net input.) The overall association (net input) reflected the degree to which different configurations of symptoms were associated with a particular disease. It was only *after* learning was completed that Gluck and Bower used the logistic function to translate net input into a judgment about the likelihood that a particular disease was associated with a particular configuration of symptoms. The logistic function was never used to generate such responses during training.

8.5.3 Tacit Theories of Responding

Some knowledge is tacit: we can know more than we can tell (Polanyi, 1966). The Rescorla-Wagner model presents an interesting variant of this theme, where if there is no need for a response theory, then there is no need to specify it. It describes how associations can be modified without the need for explicitly stating a theory of behavior. The theory of responding can be tacitly assumed until the point in time when there is a need to convert associations into responses. Interestingly, evaluations of the Rescorla-Wagner do not go beyond such a tacit assumption about behavior (Miller et al., 1995; Walkenbach & Haddad, 1980). Researchers can evaluate the Rescorla-Wagner model by agreeing that associations will eventually lead to responses, without actually stating how this is done. In the Rescorla-Wagner model, learning comes first, and responding comes later -- maybe.

This is the key difference between the Rescorla-Wagner model and the perceptron. In the perceptrons presented in earlier chapters, associations are only modified after they have produced a response. The perceptron paradox arises because, in the perceptron, response theories are not tacit. In the perceptron, there is no learning without responding.

8.6 The Impact of Responding While Learning

It might seem that the difference between "responding after learning" and "responding while learning" involves purely semantic details. However, this is not the case. It is this difference that is responsible for the empirical discrepancies between the Rescorla-Wagner model and the integration device that were reported in Chapter 7. To see why this is so, let us consider one phenomenon that was discussed in that chapter, the overexpectation effect.

8.6.1 Overexpectation

Recall from Section 7.7 that the overexpectation effect is produced with a simple two-phase methodology. In Phase 1, an agent learns to respond maximally to CS_A and CS_B by having each stimulus independently paired with a US. In Phase 2, CS_A and CS_B are presented simultaneously as a compound stimulus that is paired with the US. Overexpectation is the counterintuitive effect in which this second phase of training reduces responses to CS_A and CS_B when they are presented later as individual stimuli (Blaisdell et al., 2001; Khallad & Moore, 1996; Kremer, 1978; Lattal & Nakajima, 1998; Rescorla, 1970, 1999).

8.6.2 Learning without Responding

Because the Rescorla-Wagner model can modify associations without requiring them to be translated into responses, it provides accounts for the overexpectation effect. Let us assume that the maximum associative strength in the Rescorla-Wagner equation (λ) is equal to 1. In the first phase of the overexpectation paradigm, CS_A and CS_B are presented separately. At the end of Phase 1, the associative strengths for both CS_A and CS_B will have achieved a value of 1 (i.e., a value equal to λ).

In the second phase of the overexpectation paradigm, CS_A and CS_B are presented simultaneously. When this phase begins, the associative strength for each stimulus is equal to 1. Therefore, the overall associative strength for the compound stimulus is equal to 2. However, this is larger than the value of λ, which is still equal to 1 in Phase 2 of the paradigm. As a result, the term (λ - ΣV) in the Resorla-Wagner equation will be equal to -1 in this example. Because this value is negative, the associative strengths for both CS_A and CS_B will decrease. This decrease produces the overexpectation effect.

Note several characteristics of this account. First, the associative changes that have been described have not required responses. Second, this does not preclude the possibility of converting the existing associative strengths into responses at any point in the paradigm. Third, as was noted earlier, if the associations were converted into responses, then this would likely be accomplished by assuming a monotonic relationship between ΣV and response intensity.

8.6.3 Learning with Responding

Importantly, if a monotonic relationship exists between ΣV and response intensity, and if associative changes are contingent upon responses, then the overexpectation effect is not produced. This is why this effect is not observed when the integration device is trained (Dawson & Spetch, 2005).

Assume that after Phase 1, both connection weights from CS_A and CS_B are equal to 1, and that a net input of 1 is sufficient to turn the output unit of an integration device on. In Phase 2, both input units are activated, which produces a net input equal to 2. However, now associative changes are contingent upon the correctness of the behavior. The monotonic relationship between net input and the output unit's response dictates that if a net input of 1 turns the unit on, then so too will a net input of 2. Therefore the integration device will correctly respond to the compound stimulus. As a result, connection weights will not be modified, and the response of the integration device to the component stimuli will not change. In short, because learning in the integration device is contingent on responding, the overexpectation effect is not produced. That the effect is produced by the Rescorla-Wagner equation is because it defines learning that does not depend upon responding.

8.7 The Impact of Responding while Forgetting

8.7.1 Facilitated Reacquisition

Recall from Section 7.2 a paradigm in which an association is learned, then extinguished, and finally learned again. Facilitated reacquisition occurs when the second learning of the association is accomplished faster than was the original conditioning (Napier et al., 1992). In the memory literature, this sort of phenomenon is frequently called savings (Roediger, 1990).

8.7.2 Forgetting without Responding

We saw previously in Chapter 7 that one of the failures of the Rescorla-Wagner model was its inability to predict facilitated reacquisition (Miller et al., 1995). The reason for this failure is the model's modification of associations in the absence of responses.

In the Rescorla-Wagner model, the associative strengths for any CS would be equal to 0 prior to learning, indicating a complete absence of knowledge. When associations are formed in the Phase 1 of a facilitated acquisition study, they would eventually reach a maximum value of λ. Phase 2 of the study, extinction, is designed to make the system forget whatever associations were formed in Phase 1. However, for the Rescorla-Wagner model extinguishing the associations is identical to driving their values back down to 0. As this was their state at the start of the study, Phase 3 – reacquisition – should require the same number trials as did Phase 1. As a result, savings are not predicted.

8.7.3 Forgetting with Responding

In contrast, we saw in Section 7.2 that an integration device will produce facilitated reacquisition. The reason for this is because for this system, both learning and forgetting are contingent on network responses.

In Phase 1 of a study, the weights of a perceptron are usually initialized to be small, random numbers. The purpose of learning is not to increase connection weights up to a value of λ, but is instead to make these weights large enough to produce net inputs that cause the output unit to turn on. In Phase 2, the purpose of extinction is not to make the weights equal to zero. Rather, it is to decrease to the weights to the point that the output unit has "forgotten" the Phase 1 learning – that is, the output unit fails to respond to the stimuli. Importantly, the output unit will fail to respond when the connection weights are sufficiently small. However, they may still be dramatically different from the weight values that the network began with in Phase 1. As a result, the network demonstrates savings.

In short, the integration device produces savings because it defines extinction as "failure to respond"; the Rescorla-Wagner model does not produce savings because it defines extinction as "zero strength associations".

8.7.4 The Implications of Savings

One important aside to make at this point is that the demonstration of savings in a neural network is an important response to a common criticism of simulation approaches to animal learning.

Catastrophic forgetting occurs when the learning of new patterns suddenly and completely removes knowledge due to previous learning, and is a concern for neural network modelers (French, 1999; Robins, 1995). Furthermore, the fact that networks might exhibit catastrophic forgetting has been used to argue that networks may not be appropriate for the study of animal learning (Pearce, 1997).

The existence of catastrophic forgetting depends upon how it is measured. If one measures the degree of forgetting in networks by examining how many trials are required to reestablish previous learning, then catastrophic forgetting in networks is not as severe a problem as some would believe (Hetherington & Seidenberg, 1989). Indeed, if catastrophic forgetting is characteristic of networks, then we would not expect integration devices to predict facilitated reacquisition.

8.8 Paradox Lost

8.8.1 Computational Equivalence

The computational level of analysis is a level at which one answers the question "what information processing problem is a system solving?," and then proceeds to answer this question by performing some formal or mathematical analysis (Dawson, 1998; Marr, 1982; Pylyshyn, 1984). It is at this level of analysis that there is strong relationship between artificial neural networks and the Rescorla-Wagner equation.

It has long been known that distributed associative memories, which can be viewed as being perceptrons with linear activation functions, are formally equivalent to the Rescorla-Wagner model. This is because one can take the delta rule for training such a network and translate it into an equation that is identical to that of Rescorla and Wagner (Gluck & Bower, 1988; Gluck & Myers, 2001; Sutton & Barto, 1981).

In Chapter 5, we were able to extend this analysis into true perceptrons – that is, networks that employ nonlinear activation functions. Again, a learning rule for such a network was translated into the Rescorla-Wagner equation.

The ability to translate a rule from one domain (network learning) into another (animal learning) provides a proof that the particular rules are statements of the same information processing problem. What is this problem? Perhaps the best informal statement of it is provided by Rescorla and Wagner (1972, p. 75): "Certain expectations are built up about the events following a stimulus complex; expectations initiated by that complex and its component stimuli are then only modified when consequent events disagree with the composite expectation."

8.8.2 Algorithmic Nonequivalence

At the algorithmic level of analysis, one determines what specific processing steps are used to solve an information processing problem of interest. While the computational level of analysis demands the use of formal methods, the algorithmic level is much more concrete, demanding empirical and behavioral research practices (Dawson, 1998). Importantly, for any single computational problem, there exist an infinite number of different algorithms for the problem's solution (Johnson-Laird, 1983).

The formal equivalence of the delta rule and the Rescorla-Wagner model at the computational level hides crucial algorithmic differences between these two approaches. That is, while both are equivalent at an abstract level – both modify weights when "expectations" are false – the specific algorithms for modifying weights are fundamentally different. The "perceptron paradox" is due to these algorithmic differences, which exist in spite of the fact that the two approaches are computationally equivalent.

The fundamental algorithmic difference between the two concerns whether a response is required before associations are modified. The Rescorla-Wagner model assumes that learning mechanisms have immediate access to associative variables (λ, ΣV), and can therefore directly modify associations. In contrast, the delta rule and its variants assume that before associations are modified, they must first be converted into responses. The "expectation" that drives subsequent associative changes is the correctness of these responses. Therefore the delta rule does not have immediate access to associations – it modifies them indirectly, after they have been "filtered" into output unit responses.

In many cases, this algorithmic difference is not consequential. We have seen many examples of the agreement between machine learning and animal learning. However, in some situations this algorithmic difference is crucial, and actually leads to a disagreement between network models and animal learning models. Several examples of such disagreements were presented in Chapter 7. The implications of the computational and algorithmic relationships between the two approaches are the topic of the next and final chapter of this book.

Chapter 9: Perceptrons, Associations, and Implications

The purpose of this final chapter is to consider the implications of the formal and empirical results that have been presented in previous chapters. First, it points out the crucial fact that two systems that are equivalent at the computational level need not be equivalent at the algorithmic level. For this reason, formal analyses and computer simulations are both required for the study of associative learning. The chapter then proceeds to examine a number of specific issues that emerge from this. Some of these are related to computational issues, while others are related to algorithmic topics that emerge when one converts a formal theory into a working computer program.

A peer-reviewed monograph published by Comparative Cognition and Behavior Reviews on behalf of the Comparative Cognition Society.

9.1 Levels of Analysis and of Equivalence

9.1.1 Levels of Analysis

In cognitive science, researchers frequently structure their research programs around different levels of analysis (Dawson, 1998; Fodor, 1975; Marr, 1982; Pylyshyn, 1984). Computational analyses provide formal accounts of what information processing problems are being solved by agents. Algorithmic analyses provide empirical evidence concerning the specific processing steps that are required to solve these problems. Implementational analyses investigate the mechanisms, usually biological, that implement these different processing steps.

The view that an account of information processing requires all three different levels of analysis is called the tri-level hypothesis. It has been argued that the tri-level hypothesis can be usefully applied to the study of associative learning (Shanks, 1995).

9.1.2 Levels of Equivalence

One of the consequences of the tri-level hypothesis is that two different systems can be equivalent at one level of analysis, but can be different at another. For example, two systems are considered weakly equivalent if they are solving the same information processing problem (computationally equivalent), but are doing so by employing completely different algorithms (Pylyshyn, 1984).

One consequence of weak equivalence is that even though the two systems are solving the same problem, the fact that they are doing so using different algorithms implies that there will be observable differences in their behavior. Computational equivalence does not guarantee behavioral equivalence.

Consider, for example, a chess-playing computer program and a human chess player. If we limit our inquiry to the domain of chess, these two systems are computationally equivalent. That is, they are both systems that adhere to the rules of chess, and that share the same goal of beating their opponent. However, they have marked differences from one another at the other levels of analysis.

For instance, the chess playing program Deep Thought (Hsu, Anantharaman, Campbell, & Nowatzyk, 1990). It was constructed from customized hardware and specialized search algorithms that enabled it to evaluate 750,000 different chess positions per second, permitting it to search far into a game tree that describes the possible moves from current positions. In contrast, advanced human players do not appear to play by searching a game tree, but instead rely on pattern recognition to choose their next move (Simon & Schaeffer, 1992). Such dramatic differences at the algorithmic level will lead to a host of observable differences in play, ranging from which move is chosen next, to how long it takes a move to be played from a particular board position.

9.1.3 Simulation Required

The moral of the above example is that the claim that the delta rule and the Rescorla-Wagner model are computationally equivalent (Sutton & Barto, 1981) does not guarantee that both will generate the same responses.

As a result, one still needs to observe the responses of machine learners to identify similarities and differences between them and animal learners – regardless of their formal relatedness. If one uses computer simulations to study models of associative learning, then one can discover surprising new phenomena that go beyond our formal understanding. For instance, there are several demonstrations in Chapter 7 that when an integration device is trained, it can generate results that count as correcting failures of the Rescorla-Wagner model (Miller et al., 1995). To generate these results, one has to ignore computational equivalence, and explore the computer simulation's performance in a variety of situations.

9.2 Artificial Neural Networks and the Implementational Level

9.2.1 Applying the Three Levels

As was noted in the preceding section, it has been argued that the tri-level hypothesis can, and should, be applied to the study of associative learning (Shanks, 1995). "Associative learning can best be understood by considering three questions. What does the system do? How, in broad informational terms, does it do it? And how is this achieved at the mechanistic level?" (p. 104).

However, Shanks' (1995) application of the tri-level hypothesis provides an awkward link between associative learning and artificial neural networks. Shanks argues that contingency theory provides a computational account of associative learning. He then notes that the memorization of instances provides an algorithmic or representational account of associative learning. Finally, Shanks claims that artificial neural networks provide an implementational account of associative learning.

9.2.2 Re-evaluating the Application

There is no doubt that the tri-level hypothesis has a role to play in the study of associative learning. However, Shanks' (1995) particular account of its application seems flawed. In particular, it is much more likely that artificial neural networks have important contributions to make at the computational and algorithmic levels, and probably have much less to contribute to an implementational account of this type of learning.

To make a case, let us begin with a consideration of the implementational contributions that researchers believe can be made by perceptrons, which we have shown throughout this book to have a strong link to studies of classical conditioning. To what extent do researchers believe that this link is implementational in nature?

It is certainly true that early developments in artificial neural networks were driven by implementational goals. As was noted in Chapter 2, the McCulloch-Pitts neuron was developed in an attempt provide a logical account of brain activity (McCulloch & Pitts, 1943). The same is true of the perceptron. Rosenblatt (p. 3) viewed perceptrons as being "brain models": "By 'brain model' we shall mean any theoretical system which attempts to explain the psychological functioning of a brain in terms of known laws of physics and mathematics, and known facts of neuroanatomy and physiology" (Rosenblatt, 1962).

However, by the end of his seminal work on perceptrons, Rosenblatt (1962) was cautious in his interpretation of the relation of his models to the brain. While he was attempting to build "brain models", he realized their limitations. "At present, any treatment of the compatibility of perceptron theory with biological memory mechanisms must remain entirely speculative" (p. 570).

Many modern treatments of the relation between models of the type that we have discussed and the brain have converted this caution into criticism. It is possible to generate long lists of properties that are true of artificial neural networks but are not true of the brain (Crick & Asanuma, 1986; Smolensky, 1988). Artificial networks have been ridiculed as being "stick and ball" models of the brain (Douglas & Martin, 1991). In short, many would argue that implementational contributions will be the smallest to be made by artificial neural networks.

Nonetheless, these networks can make contributions. Interestingly, these seem to be at the other two levels that are part of the tri-level hypothesis. This book has provided many examples of the computational relationship between animal learning and machine learning. This book has also provided many examples of the algorithmic relationship between the two. To the extent that the various perceptrons that have been encountered in this book have provided interesting insights into associative learning or classical conditioning, these insights have been at the two levels that Shanks (1995) suggests do not involve artificial neural networks.

9.3 Perceptrons and Theories of Everything

9.3.1 Theory on a T-shirt

Modern physicists have a dream of producing a "theory of everything"; an elegant mathematical statement from which the nature of the physical world follows. "My ambition is to live to see all of physics reduced to a formula so elegant and simple that it will fit easily on the front of a T-shirt" (Lederman, 1993, p. 21).

Quantitative approaches to psychology have long been inspired by physics, as (Köhler, 1947) noted in his famous critique of behaviorism. "Since the enormous feat of jumping from the world of direct, but confused, experience into a world of clear and hard reality already has been achieved by the physicist, it would seem wise for the psychologist to take advantage of this great event in the history of science, and to begin the study of psychology on the same solid basis" (p. 5). Modern mathematical psychologists are certainly hopeful of the discovery of psychological laws that are as valid as the laws of physics (Estes, 1975). However, few would be expecting to find a psychological theory of everything to put on a T-shirt.

However, that isn't to say that mathematical psychologists do not have a preference for simple but powerful quantitative statements. Modern mathematical psychologists evaluate their theories by seeking a trade-off between simplicity and power: a simple theory that provides good fit to data, and which generalizes to new data, is to be preferred (Pitt, Myung, & Zhang, 2002). One of the appeals of the Rescorla-Wagner model is that it possesses such properties (Blaisdell, 2003).

9.3.2 The Poor Perceptron

Such considerations lead to an obvious problem with perceptron models of associative learning. Perceptrons have been subjected to an intensive computational analysis (Minsky & Papert, 1988). After endorsing the computational role of perceptrons in Section 9.2, this analysis becomes fair game. And what it reveals is that there are many input-output mappings that cannot be represented by a perceptron. In short, perceptrons provide simple and elegant theories. However, there are good reasons to be concerned about their limited power.

We have already seen that perceptrons can only represent solutions to problems that are linearly separable. For models of associative learning, this restriction is severely limiting (Quinlan, 1991). As a result, when most modern researchers explore artificial neural networks as models of associative learning, they do so by studying multilayer networks (Enquist & Ghirlanda, 2005). Such networks have extraordinary computational power (Siegelmann, 1999), and in principle should be capable of modeling any stimulus-response regularity (McClelland et al., 1986).

9.3.3 A Quilt of Small Theories

The above-mentioned limitations of perceptrons, however, are only a concern if one is seeking a single theory of associative learning that is analogous to a "theory of everything". There are some who believe that psychology is not well-suited to this theoretical goal. "In physics, we're used to explanations in terms of perhaps a dozen basic principles. For psychology, our explanations will have to combine hundreds of smaller theories" (Minsky, 1985).

Shanks (1995, p. 127) notes "people have no difficulty learning nonlinearly-separable classifications which the delta rule model we have been considering would be unable to master". However, such a critique should only be used to abandon perceptrons in the search of a single unifying theory. We have seen that different types of perceptrons capture different sets of empirical regularities. Perceptrons (including variations that use different activation functions) could provide a set of smaller theories – contributing one or more theoretical squares to psychology's quilt. That is, while perceptrons are clearly not powerful enough to be theories of everything, they are powerful enough to serve as theories of a potentially large part of associative learning. Their computational limits do not necessarily translate into their demise.

9.4 Guilt by Association

9.4.1 Another Computational Problem

Computational considerations lead to another problem with viewing perceptrons as interesting models for the study of associative learning. The computational equivalence between a learning rule for perceptrons and the Rescorla-Wagner model means that perceptrons should inherit all of the advantages that the Rescorla-Wagner model enjoys. However, it also means that perceptrons should inherit the disadvantages of the Rescorla-Wagner formulations, of which there are many (Miller et al., 1995). If these problems are serious enough, then this might be grounds to abandon perceptron models -- guilt by association.

This is certainly the perspective taken by some learning researchers. For example, one perceptron-like model has been proposed to account for associative learning involving configural cues (Gluck, 1991), but has been critiqued because of its formal equivalence to the Rescorla-Wagner model: "Because it is formally equivalent to the Rescorla-Wagner model, it makes the same erroneous predictions" (Pearce, 1997, p. 132). Similarly, Schmajuk has proposed neural network models of associative learning that are related to the perceptrons described earlier, but far more ambitious because they attempt to incorporate temporal variables that are known to affect conditioning (Schmajuk, 1997). This research has also been critiqued because it is viewed as having the same problems as the Rescorla-Wagner model (Blaisdell, 2003).

9.4.2 The Learning-Performance Distinction

The problem with assuming that the formal equivalence between two learning rules means that both rules will have the same limitations is the failure to recognize that such computational equivalence does not imply equivalence at other levels of analysis. Indeed, if two systems are computationally equivalent, but are not algorithmically equivalent, then they can generate different behaviors. The critiques cited above fail to take this into consideration.

That this is a problem was revealed in the empirical results that were presented in Chapter 7. In that chapter, we considered several phenomena that are documented as failures of the Rescorla-Wagner model (facilitated reacquisition, retarded reacquisition, extinction of a conditioned inhibitor, and latent inhibition). However, we also demonstrated that these phenomena can be produced in an integration device. In other words, the formal equivalence between learning in these two theories (see Chapter 5) did not prevent the integration device from providing an account of how some of the Rescorla-Wagner failures might be overcome.

The results in Chapter 7 provide sufficient evidence on their own to indicate that it is not wise to 1) consider two theories only at the computational level, and 2) dismiss a new theory on the basis of this comparison without 3) extending this consideration to other levels of analysis. By changing the algorithm that brings a particular computational regularity to life, one can produce behaviors, or account for results, that are not necessarily the sole domain of the computational regularity.

To my mind, this is one of the intriguing aspects of one of the challenges to modern associative theories (Miller, 2006). It has been argued that most of these theories only provide an account of how associative strengths are modified (learning), and do not provide an account of how they are converted into responses (performance). What might be an advantage of incorporating performance into a theory of learning? The perceptrons that we have been considering provide one example of including both learning and performance, and have demonstrated that one of its unanticipated benefits is the ability to make predictions which cannot be made by the theory of learning alone. In short, before we abandon them because of computational equivalence, we might want to explore their algorithmic level advantages!

9.5 The Paralyzed Perceptron

9.5.1 Paralyzed Learning

One issue that is related to the learning-performance distinction, and which has a long history in the study of associative learning, is whether an agent must produce an overt response in order for learning to occur. Harlow and Stagner (1933) called theories in which such a response was mandatory peripheral, and theories in which such a response was not necessary intra-cerebral. They conducted a number of experiments in which cats and dogs were conditioned when under the influence of curare. Curare is a drug that paralyzes the striped musculature, preventing responses from being made, but does not affect spinal cord or midbrain receptors and reflex centers. Harlow and Stagner were unable to demonstrate learning when animals were under the influence of this drug, and took this as evidence in favor of peripheral theories of learning. "Conditioned reflexes are only possible if a response is made, and do not result merely from stimuli presented simultaneously" (p. 293).

Subsequent studies (e.g., Girden, 1940; Solomon & Turner, 1962) have challenged some of the conclusions drawn by Harlow and Stagner (1933). For example, Solomon and Turner demonstrated what when dogs were under the influence of curare they could learn to discriminate tones that were signals of subsequent shocks from tones that were never associated with shocks. In short, it has been demonstrated that associations can be formed in the absence of responses.

9.5.2 What Is A Response?

It has been argued in previous sections that the key difference between perceptron learning and the Rescorla-Wagner model is the fact that the former requires a response, while the latter does not. However, if it is possible to form associations in the absence of responding, then this might indicate a situation in which the Rescorla-Wagner formalism is more robust than the simple neural networks that we have been surveying.

There are two points to make about this issue. First, it would be a mistake to equate the activity of an output unit with the performance of some observable behavior. While it is certainly plausible to relate behavior to activity, it is not necessary. For instance, one could instead interpret output unit activity as a judgment about the likelihood of reinforcement occurring – activity as expectation -- when a particular stimulus is encountered. An example of this interpretation of activity is provided by Gluck and Bower (1988). If one were to adopt this interpretation of output unit activity, then the neural network would constitute an intra-cerebral theory.

The second point is that when a neural network is being used to simulate a particular animal experiment, then the simulator has a responsibility to state the intended interpretation of output unit activity. It could represent expectation, or it could represent observable behavior (e.g., activity = rate of performing some response). If this latter interpretation is intended, then perceptrons would not be an appropriate model for learning under the influence of curare. If activity is behavior, and behavior is required for learning, then paralyzed perceptrons would be unable to form associations.

9.6 Perceptrons and Input Representations

9.6.1 Elements vs. Configurations

What sorts of computational issues might we explore with artificial neural networks? One important issue concerns how animals process compound stimuli. One approach to this issue is elemental (Rescorla, 1973, 1988, 2003; Rescorla & Wagner, 1972; Wagner, 2003): animals independently process the elemental components that define a compound stimulus.

A second approach to this issue is configural (Pearce, 1987, 1994, 1997, 2002; Pearce & Bouton, 2001; Wasserman & Miller, 1997): compound stimuli are processed holistically: while a compound stimulus is related to its components, there is additional information that is represented that reflects the notion that a compound stimulus is not merely the sum of its elements.

9.6.2 Network Representations

Elemental and configural approaches have led to different stimulus representations in artificial neural networks. A typical elemental representation would have a single input unit representing the presence or absence of each element of a compound stimulus, and no other input units (e.g., Pearce, 1997, p. 131). For example, a simple network would use two units that could represent the presence of CS_A and CS_B. In an elemental representation, if both of these units are on, then the compound stimulus CS_{AB} is present.

In contrast, a typical configural representation includes additional units that are turned on only when a compound stimulus is presented (e.g., Pearce, 1997, p. 132). For example, three units would be required to represent compound stimulus CS_{AB}: one for element CS_A, another for element CS_B, and a third to represent unique configural properties (CS_{AB}) only when the compound is present.

9.6.3 Distributed Representations

In accepting that perceptrons can inform studies of associative learning, it becomes apparent that other forms of input representations are possible. Elemental and configural representations are local, in the sense that each input unit represents the presence or absence of a complete CS. However, this is not necessary. For example, one could use a distributed representation in which a single CS is described as a configuration of features. The entire set of features defines a particular CS.

For instance, Wagner's SOP theory (e.g. Mazur & Wagner, 1982; Vogel, Brandon & Wagner, 2003; Wagner, 1981) is designed to model the changes in concept activity, and associations between concepts, over time. It represents stimuli as a distributed population of elements that can be in different states of activity at any given time; the activity of a stimulus is a function of pooling the activities of its component elements.

One example of a distributed connectionist representation has been used in a recent study of discriminant learning in chickadees (Nickerson, Bloomfield, Dawson, & Sturdy, 2006). The chickadee call consists of different note types (A, B, C, and D). In a previous study (Charrier, Lee, Bloomfield, & Sturdy, 2005), birds were trained to respond to one note type (e.g., A+) and to not respond to another (e.g., B-). Nickerson et al. replicated this study using perceptrons. Each stimulus note was represented as a set of 9 feature values that were used to activate 9 input units of a perceptron. The perceptron was trained to respond to some notes, and not to others. Results indicated an extremely strong relationship between perceptron responses and the previously observed behavior in the birds.

Of interest in this study is that the representational form is not clearly elemental or configural. It is elemental, in the sense that each feature is a component of a note, but it is also configural, in the sense that each stimulus is defined by an entire configuration of features. At the very least, this suggests that perceptrons can provide a medium in which a wide variety of input representations can easily be informed, and the results of these simulations should contribute to the elemental/configural debate.

9.7 Perceptrons and Theories of Performance

9.7.1 Exploring Performance

The main reason that perceptrons generate predictions that differ from those of the Rescorla-Wagner model is that perceptrons focus on learning and performance, instead of emphasizing learning alone. This means that perceptrons can also be used to explore a variety of theories of performance.

This is because one can change a perceptron's "theory of performance" by changing its activation function. The role of the activation function is to convert internal associations (represented as net inputs) into observable responses. If the activation function is changed, then the mapping from stimulus to response changes as well.

This is not a trivial point to make, because the response changes that result from altering the activation function may be dramatic.

For example, we have seen that a standard critique against the use of perceptrons to model associative learning is their inability to handle linearly separable problems like XOR (Pearce, 1997; Quinlan, 1991; Shanks, 1995). However, small problems like XOR can indeed be solved by a perceptron if it uses the appropriate activation function (Dawson, 2005, Chapter 14). For example, the integration device that employs the logistic equation cannot cope with XOR. However, if the logistic is replaced with the Gaussian equation to convert the device into a value unit, then XOR is easily handled.

This is not to say that value units are uniformly more powerful than integration devices. One of the costs of changing activation functions is a network type by problem type interaction (Dawson, 2005, Chapter 14). That is, a particular activation function is excellent for some problems, but has difficulty with others. Integration devices fail on XOR, but have no trouble with OR. The reverse is true for value units.

However, irrespective of the issue of power, the logical capabilities of a device depend enormously upon the activation function, independent of the learning rule. The activation function is also a crucial component for governing the responses of a system. We saw in Chapter 7 that an integration device will not produce the overexpectation effect. However, this effect is characteristic of a value unit.

9.7.2 Activation Implications

Previous chapters have illustrated that many of the presumed failures of the Rescorla-Wagner model might be dealt with by supplementing it with a specific performance theory. Three such theories have been presented in this book, in the form of three different activation functions for perceptrons: the Heaviside equation, the logistic equation, and the Gaussian equation. It is important to realize that these three represent the small tip of a very large iceberg of theories that is crying out for further study.

In a recent survey of artificial neural networks, it was determined that at least 640 different activation functions have appeared in the literature (Duch & Jankowski, 1999). This diversity represents a wealth of alternative theories of performance that could, and probably should, be studied in the context of associative learning.

In addition, different activation functions are often accompanied by different learning rules and updated learning algorithms. What is the formal relationship between these learning rules and the Rescorla-Wagner model? Might these different activation functions and learning rules be more related to other models that have emerged in an attempt to deal with some of the problems that the Rescorla-Wagner model has encountered (Mackintosh, 1975; Pearce & Hall, 1980)? Miller (2006) has called for renewed emphasis on reconciling the distinction between learning and performance. The variety of possible perceptron architectures provides a fruitful domain for this issue to be investigated.

9.8 Beyond the Perceptron

9.8.1 Perceptron Limitations

Hopefully the many simulation results that have been presented in this book have shown that a very simple type of artificial neural network is capable of performing a surprisingly large number of associative learning tasks. Nonetheless, the simple structure of these networks limits their capabilities. It is fruitful to ask the question "How much can one do with a perceptron?", and one goal of this book has been to attempt to provide an answer to this question. However, it would be foolish not to acknowledge that perceptrons must fail to model all of the results in the extensive literature on classical conditioning.

As was noted in Sections 1.6 and 4.16, the models that we have discussed are not capable of dealing with important temporal issues that govern conditioning. It is not clear how to use these networks to capture results from second-order conditioning. Many results that appear to occur in the absence of explicit learning, such as spontaneous recovery from overshadowing, cannot be handled by perceptrons.

9.8.2 Multilayer Networks

However, it is possible to develop neural network models to deal with all of these difficulties. In principle, neural networks with the appropriate properties are capable of modeling any computable function, and therefore are as powerful as any type of model available to cognitive science (see Dawson, 1998 for a detailed discussion of this issue). Several such models exist in the associative learning literature.

For example, Gibbon (1977) proposed scalar timing theory as a quantitative model of how animals estimate interval durations, count how many events have occurred, and estimate the rate at which events occur. Church and Broadbent (1990) have proposed a neural network version of scalar timing theory. The network consists of a set of oscillators of different timing cycles. Each oscillator has associated with it another processor that receives an oscillator's signal, and determines the oscillator's phase. Distributed working and reference memories are implemented as sets of connection weights. By pooling the signals of different oscillators running at different frequencies, accurate temporal judgments can be mediated.

Similarly, a variety of associative learning paradigms have been successfully modeled by networks developed by Schmajuk and his colleagues (e.g., Schmajuk, 1997; Schmajuk & Blair, 1993; Schmajuk, & DiCarlo, 1992; Schmajuk, Lam, & Gray, 1996; Schmajuk, Lamoureux, & Holland, 1998). Consider for a moment one of the earliest of these networks, Schmajuk and DiCarlo's S-D network. This consists of a set of input units that represent the presence of various CSs. These input units have direct connections to units that represent CS-US associations, which can mediate conditioned responses. However, they also have direction connections to a set of intermediate or hidden units that are capable of representing configural relations among various CSs, and which in turn provide signals for the CS-US association processors. The model is highly temporal in nature, because CS representations are time-based, and network computations are all instances of computing changes (e.g., in activity or in associations) as a function of changes in time.

Finally, a number of studies have simulated associative learning experiments by training multilayer perceptrons in a fashion very similar to the simulations that have been described in this paper (e.g., Delamater, Sosa & Katz, 1999; Nickerson et al., 2006; Yaremchuk et al., 2005). These networks – as well as those cited above in this section – differ from the perceptrons that we have discussed in virtue of having hidden units. These hidden units permit nonlinear relationships among input units to be captured (for a discussion see Dawson, 2004), which in turn permit them to solve problems whose solutions are impossible to represent in a perceptron that does not have hidden units (compare Rumelhart & McClelland, 1986 to Minsky & Papert, 1969).

9.9 Formalism, Empiricism, and Multilayer Models

9.9.1 A Simple Lesson

Given the existence of the more powerful networks that were briefly discussed in Section 9.8, one might question the worth of this book's exploration of the properties of much simpler systems. What is to be learned from algorithmic and formal analyses of models that are so dated?

One lesson that our examination has provided is that a complete understanding of a model, and of its relationship to other theories, requires us to answer both algorithmic and computational questions. If one focuses on only one type of question, fundamentally important regularities will be lost.

9.9.2 Multilayer Implications

This simple lesson is important to keep in mind when studying associative learning with models that are more powerful than the perceptron. In psychological modeling, there is a strong tendency to view every model as a model of data, and to evaluate it in terms of its fit to empirical data (e.g., Dawson, 2004). When a model fails to capture regularities, the urge is to quickly patch it or replace it with a model that will perform better (Dawson & Shamanski, 1994). However, such an algorithmic emphasis ignores crucial, and interesting, formal questions that must also be faced.

For instance, Weisman et al. (1999) tested the ability of Church and Broadbent's (1990) connectionist model of timing, and found that it was unable to generate judgments of the type "a signal is longer than a duration of *x*, but shorter than a duration of *y*". Clearly if the model is to fit such data then it must be changed. But how should it be changed? One fruitful direction to take would be to conduct a formal analysis of the network in an attempt to understand why such judgments were beyond it. At first glance, this inability seems analogous to a standard perceptron's inability to perform negative patterning (see Section 6.10). Is it possible that the layers of units in Church and Broadbent's model are not capturing configural relationships amongst inputs, and are therefore limiting its computational power?

As another example, in Section 6.4 it was pointed out that from an algorithmic perspective having negative biases in a perceptron may not be desirable. However, if bias is restricted (e.g. if it is forced to remain positive during training, or is fixed during training), then this further restricts what problems that a perceptron can solve. This is because this restricts the position of the "cut" in pattern space made by the output unit (e.g. Section 6.8.2). Other models, such as Miller and Matzel's (1988) comparator process do not propose inhibitory associations. While formalization of the comparator hypothesis has not yet been completed (e.g. Savastano et al., 2003), it is important to determine whether the absence of inhibition in it restricts the kinds of problems that it can solve, as it would in a perceptron.

Finally, Shanks (1995) has argued that contingency theory (e.g., Cheng & Holyoak, 1995) provides a complete computational theory of associative learning. He reviews formal evidence that establishes that the Rescorla-Wagner model of learning is formally equivalent to contingency theory, and goes on to argue that artificial neural networks simply provide an implementational or biological account of how such theories are physically realized. However, it is well known that multilayer networks are far more powerful than perceptrons (e.g. Rumelhart & McClelland, 1986). If multilayer networks are ultimately required to completely describe the results of learning experiments, and if they are more powerful than either perceptrons or the Rescorla-Wagner model, then it must also be the case that they provide a formalism for associative learning that is more powerful than contingency theory. What is this formalism? In what ways does it represent advances over contingency theory? What are the implications of this for accounts of associative learning? These important questions can only be answered by following the simple lesson offered by the perceptron: by combining formal and algorithmic research methods.

9.10 From Informal to Formal, and on to Workable

9.10.1 Formal Advantages

"Theory in a field as immature as psychology cannot be expected to amount to much -- and it doesn't" (Royce, 1970). What can be done to improve psychological theories? One possibility is to translate an informal verbal theory into a formal mathematical expression

There are several reasons that the process of formalization is useful (Lewandowsky, 1993). First, it adds precision in specifying theoretical terms. An informal theory can be full of references to terms with vague definitions like "memory" or "attention". Many academic debates emerge because different researchers use the same terms in different ways. In a formal model, conceptual terms have to be carefully operationalized in order for the model to work.

A second advantage of formalization is that the language in which a theory is expressed determines the kinds of ways in which the theory can be tested or explored. For instance, after a verbal theory has been formalized mathematically, one can use mathematical operations to investigate its implications (Coombs, Dawes, & Tversky, 1970; Lunneborg, 1994; Wickens, 1982). In other words, formalization not only results in a more precise specification of the concepts in the theory, but also results in a more precise set of tools for studying these concepts.

A third advantage of formalization is that it can reveal hidden assumptions in an informal theory which themselves need to be fleshed out in greater detail in order for the theory to be complete. For example, many theories in cognitive psychology are expressed as flowcharts of black boxes. Ideally, each black box in such a flowchart is supposed to be a primitive operation that needs no further explanation (Cummins, 1983; Dawson, 1998). A formal statement of a theory, coupled with a formal analysis of its concepts, can reveal that some of these alleged primitives are themselves very complicated processes that require further analysis and explanation.

9.10.2 From the Formal to the Workable

Interestingly, the mere formalization of a theory may only be a first step in rigorously understanding it. The previous advantages of formalization become amplified by converting a formal theory into a working computer simulation. "Even deceptively simple models can benefit from the rigor of simulations" (Lewandowsky, 1993, p. 236).

The reason for this is that the act of formalization establishes rigor at the computational level of analysis, but not at others. Converting a formal model into a computer simulation brings the algorithmic level into play. The advantages of precision, new methods of testing, and revealing hidden assumptions are not only found at the computational level, but at the algorithmic level as well.

We have already encountered specific examples of this with respect to the relationships between perceptrons and models of animal learning. At the computational level, some researchers might dismiss perceptrons because of their formal relationship to the Rescorla-Wagner model, and the presumed inheritance of Rescorla-Wagner problems (Blaisdell, 2003; Pearce, 1997). However, Chapter 7 revealed that if one goes to the trouble to take a working perceptron and train it in a number of animal learning paradigms, it actually generates results that diverge from Rescorla-Wagner model predictions. Clearly the act of simulating the perceptron generated knowledge that was not evident with only a computational or formal understanding of the models.

Further to this, once such discrepancies are uncovered through the act of computer simulation, it is imperative to take a closer look at the models to understand why they occur. The relevance of a theory of performance to the predictions made by the Rescorla-Wagner model is completely obscured without converting formal accounts into simulations, observing performance, and explaining this (algorithmic) data.

9.11 Models and Metaphors

9.11.1 Metaphor and Understanding

Cognitive scientists have long been interested in the role that metaphor plays in our understanding of the world (Ortony, 1979). A metaphor is generally considered to have two different subjects, the topic and the vehicle. The topic is a subject that we wish to learn more about, while the vehicle is a subject (or possibly an entire system of ideas) with which we are very familiar. A metaphor works by projecting our knowledge of the vehicle onto the topic, with the result that we come to have a new understanding of it. This notion of "projecting" is often assumed to be an interaction, in the sense that the fit of the vehicle to the topic depends upon using the characteristics of the topic to emphasize particular aspects of the vehicle that are going to be projected (Black, 1979).

The advantages of simulation that were described in Section 9.7 would be true even if one merely created perceptron simulations, and never attempted to relate these simulations to the domain of animal learning. It is important to realize that the simulations that have been presented in this book have attempted to bridge two different research domains, machine learning and animal learning. As a result, they offer additional advantages, because – as is the case in metaphor – they offer the possibility of seeing one domain in a new light that the other domain provides.

9.11.2 The Null Condition

Metaphoric understanding emerges when we, via computer simulation, project the animal learning domain onto the machine learning. For example, one issue that was raised in Chapter 6 was the role of the null condition. Recall that the null condition is a network training pattern in which no stimuli are presented, and the network is trained to not respond.

The need for the null condition becomes an issue when computer simulation forces us to consider the precise nature of the training set for a network, because this training set is a model of an experimental paradigm. How should the network be treated when no CSs are present?

One approach is to say that the network should not be treated at all, which means that the null condition should not be included in the training set. Some researchers have adopted this view (Delamater et al., 1999), and it may be perfectly appropriate. However, one implication of this is that the logical nature of the training sets is affected. In particular, with no null condition, negative patterning is not linearly nonseparable, as it is not logically equivalent to XOR.

An alternative approach is to say that when animals are not reinforced in the absence of stimuli, they *are* learning something (i.e., learning not to respond to a particular stimulus configuration). According to this view, the null condition should be included. Of course, this has a logical impact on the training set.

More importantly, the inclusion of the null condition leads to additional metaphoric understanding of the network. When the null condition is included, we saw enhanced effects in paradigms like overshadowing. How did these enhanced effects emerge? When interpreting network structure, it was apparent that the key role of the null condition was to alter the threshold or bias of the perceptron's output unit.

From this observation, we were then led to a completely novel interpretation of the role of the threshold or bias in a perceptron. That is, we saw that one could view the value of the bias as being the strength of the association to background stimuli. This is a completely new perspective on the perceptron, and it arises because of the need to precisely map conditions of an animal learning experiment into a working computer simulation. Thus, the computer simulation approach that has been explored in this book has provided novel metaphoric insight as well.

9.12 The Synthetic Approach

9.12.1 Analysis and Synthesis

The vast majority of research in psychology and cognitive science follows an analytic methodology (Braitenberg, 1984; Dawson, 2004). In general, it proceeds by taking an intact system, making observations of its behavior in a variety of stimulus situations, and inferring its internal rules or mechanisms on the basis of these observations. Models of behavior are produced after the behavior has first been observed.

The analytic approach has been enormously successful. However, it is important to realize that alternative approaches are also available, and that these approaches can also provide important insights into the systems that we study.

For example, one alternative research strategy is to make some assumptions about primitive capacities, and then to build these capacities into working systems or models without first analyzing intact agents. Model construction can precede empirical analysis. This approach is fundamental to research in behavior-based robotics (e.g. Brazeal, 2002; Brooks, 1999; Webb & Consi, 2001). "Only about 1 in 20 [students] 'gets it' -- that is, the idea of thinking about psychological problems by inventing mechanisms for them and then trying to see what they can and cannot do" (Minsky, 1995, personal communication).

Braitenberg (1984) has called this alternative approach synthetic psychology, and has argued that it should be adopted because theories that are derived via analysis are inevitably more complicated than is necessary. This has also been called the frame-of-reference problem (Pfeifer & Scheier, 1999). A consequence of the frame-of-reference problem is that because of nonlinear interactions (such as feedback between components, and between a system and its environment), relatively simple systems can surprise us, and generate far more complicated behavior than we might expect. The further appeal of the synthetic approach comes from the belief that if we have constructed this simple system, we should be in a very good position to propose a simpler explanation of its complicated behavior. In particular, we should be in a better position than would be the case if we started with the behavior, and attempted to analyze it in order to understand the workings an agent's internal mechanisms.

Many of the simulation results presented earlier illustrate the synthetic approach in action. First, one creates some "building blocks" from which a neural network is constructed: particular processing units, the connection weights, and a learning rule. Second, one trains the network to perform some task. The point of this training is not to fit particular data, but is rather to observe the responses of the network. Can it learn the task? If so, what are the properties of this learning? The hope in this exploration is to discover some surprising or counterintuitive regularities.

We saw in Chapter 7 that the integration device could easily generate surprising results. In general, these surprises were of the same sort: the network generated data that was contrary to the predictions of the Rescorla-Wagner model. This was in spite of the fact that the learning rule for the networks was formally equivalent to the Rescorla-Wagner equation.

The main implication of these surprises was a reevaluation of the relationship between network learning and animal learning. In general, we were reminded of the fact that computational equivalence does not equate to algorithmic equivalence. More specifically, we saw that a more careful consideration of the learning-performance distinction, with an emphasis on how associations may be converted into responses, has important consequences regarding putative successes and failures of the Rescorla-Wagner model. To the extent that the reader has found this to be surprising or interesting, it is important to realize that these issues only emerged after using the synthetic approach to study Pavlovian conditioning.

Cited Literature

Agre, P. (1997). *Computation and Human Experience*. New York: Cambridge University Press.

Anderson, J. A., & Rosenfeld, E. (1998). *Talking nets: An Oral History Of Neural Networks*. Cambridge, MA: MIT Press.

Arbib, M. A. (1964). *Brains, Machines, And Mathematics*. New York, NY: McGraw-Hill.

Ashby, W.R. (1960). *Design for a Brain, Second Edition*. New York: John Wiley and Sons.

Baker, A. G. (1974). Conditioned inhibition is not the symmetrical opposite of conditioned excitation: A test of the Rescorla-Wagner model. *Learning and Motivation, 5*, 369-379.

Ballard, D. (1986). Cortical structures and parallel processing: Structure and function. *The Behavioral And Brain Sciences, 9*, 67-120.

Balsam, P. D. (1985). The functions of context in learning and performance. In P. D. Balsam & A. Tomie (Eds.), *Context and Learning* (pp. 1-21). Hillsdale, NJ: Lawrence Erlbaum Associates.

Bechtel, W. (1985). Contemporary connectionism: Are the new parallel distributed processing models cognitive or associationist? *Behaviorism, 13*, 53-61.

Bechtel, W., & Abrahamsen, A. (1991). *Connectionism And The Mind*. Cambridge, MA: Basil Blackwell.

Bellingham, W. P., & Gillette, K. (1981). Attenuation of overshadowing as a function of non-differential compound conditioning trials. *Bulletin of the Psychonomic Society, 18*, 218-220.

Bever, T. G., Fodor, J. A., & Garrett, M. (1968). A formal limitation of associationism. In T. R. Dixon & D. L. Horton (Eds.), *Verbal Behavior And General Behavior Theory* (pp. 582-585). Englewood Cliffs, NJ: Prentice-Hall.

Black, M. (1979). More about metaphor. In A. Ortony (Ed.), *Metaphor and Thought* (pp. 19-43). Cambridge: Cambridge University Press.

Blaisdell, A. P. (2003). Proteus caught in a (neural) net. Animal learning and cognition: A neural network approach. *Journal of Mathematical Psychology, 47*(2), 229-234.

Blaisdell, A. P., Denniston, J. C., & Miller, R. R. (2001). Recovery from the overexpectation effect: Contrasting performance-focused and acquisition-focused models of retrospective revaluation. *Animal Learning & Behavior, 29*(4), 367-380.

Bouton, M. E. (1986). Slow reacquisition following the extinction of conditioned suppression. *Learning and Motivation, 17*(1), 1-15.

Bouton, M. E., & Bolles, R. C. (1979). Role of conditioned contextual stimuli in reinstatement of extinguished fear. *Journal of Experimental Psychology-Animal Behavior Processes, 5*(4), 368-378.

Bouton, M.E., & King, D.A. (1983). Contextual control of the extinction of conditioned fear: Tests for the associative value of the context. *Journal of Experimental Psychology: Animal Behavior Processes, 9*, 248-265.

Braitenberg, V. (1984). *Vehicles: Explorations In Synthetic Psychology*. Cambridge, MA: MIT Press.

Breazeal, C. (2002). *Designing Sociable Robots*. Cambridge, MA: MIT Press.

Brooks, R. (1999). Cambrian Intelligence: The Early History of the New AI. Cambridge, MA: MIT Press.

Brown, T. H. (1990). Hebbian synapses: Biophysical mechanisms and algorithms. *Annual Review of Neuroscience, 13*, 475-511.

Carpenter, G. A. (2001). Neural-network models of learning and memory: leading questions and an emerging framework. *Trends in Cognitive Sciences, 5*, 114-118.

Chang, R. C., Stout, S., & Miller, R. R. (2004). Comparing excitatory backward and forward conditioning. *Quarterly Journal of Experimental Psychology, Section B: Comparative and Physiological Psychology, 57*, 1-23.

Charrier, I., Lee, T. T. Y., Bloomfield, L. L., & Sturdy, C. B. (2005). Acoustic mechanisms of note-type perception in black-capped chickadee (Poecile atricapillus) calls. *Journal of Comparative Psychology, 119*(4), 371-380.

Cheng, P. W., & Holyoak, K. J. (1995). Complex adaptive systems as intuitive statisticians: Causality, contingency, and prediction. In H. L. Roitblat & J.-A. Meyer (Eds.), *Comparative Approaches To Cognitive Science* (pp. 271-302). Cambridge, MA: MIT Press.

Chomsky, N. (1959). A review of B.F. Skinner's Verbal Behavior. *Language, 35*, 26-58.

Church, R.M., & Broadbent, H.A. (1990). Alternative representations of time, number, and rate. *Cognition, 37*, 55-81.

Clark, A. (1997). *Being There: Putting Brain, Body, and World Together*. Cambridge, MA: MIT Press.

Clarke, J. C., Westbrook, R. F., & Irwin, J. (1979). Potentiation instead of overshadowing in the pigeon. *Behavioral and Neural Biology, 25*, 18-29.

Coombs, C. H., Dawes, R. M., & Tversky, A. (1970). *Mathematical Psychology: An Elementary Introduction*. Englewood Cliffs, NJ: Prentice-Hall.

Cotman, C. W., Monaghan, D. T., & Ganong, A. H. (1988). Excitatory amino acid neurotransmission: NMDA receptors and Hebb-type synaptic plasticity. *Annual Review of Neuroscience, 11*, 61-80.

Crick, F., & Asanuma, C. (1986). Certain aspects of the anatomy and physiology of the cerebral cortex. In J. McClelland & D. E. Rumelhart (Eds.), *Parallel Distributed Processing* (Vol. 2, pp. 333-371). Cambridge, MA: MIT Press.

Cummins, R. (1983). *The Nature Of Psychological Explanation*. Cambridge, MA.: MIT Press.

Dawson, M. R. W. (1998). *Understanding Cognitive Science*. Oxford, UK: Blackwell.

Dawson, M. R. W. (2004). *Minds And Machines : Connectionism And Psychological Modeling*. Malden, MA: Blackwell Pub.

Dawson, M. R. W. (2005). *Connectionism : a hands-on approach* (1st ed.). Oxford, UK ; Malden, MA: Blackwell Pub.

Dawson, M. R. W., Medler, D. A., & Berkeley, I. S. N. (1997). PDP networks can provide models that are not mere implementations of classical theories. *Philosophical Psychology, 10*, 25-40.

Dawson, M. R. W., Medler, D. A., McCaughan, D. B., Willson, L., & Carbonaro, M. (2000). Using extra output learning to insert a symbolic theory into a connectionist network. *Minds And Machines, 10*, 171-201.

Dawson, M. R. W., & Schopflocher, D. P. (1992). Modifying the generalized delta rule to train networks of nonmonotonic processors for pattern classification. *Connection Science, 4*, 19-31.

Dawson, M.R.W., & Shamanski, K.S. (1994). Connectionism, confusion and cognitive science. *Journal of Intelligent Systems, 4*, 215-262.

Dawson, M. R. W., & Spetch, M. L. (2005). Traditional perceptrons do not produce the overexpectation effect. *Neural Information Processing -- Letters and Reviews, 7*(1), 11-17.

De Wilde, P. (1997). *Neural Network Models, Second Edition*. London: Springer.

Delamater, A. R., Sosa, W., & Katz, M. (1999). Elemental and configural processes in patterning discrimination learning. *The Quarterly Journal Of Experimental Psychology, 52B*, 97-124.

Dickinson, A., & Burke, J. (1996). Within-compound associations mediate the retrospective revaluation of causality judgments. *Quarterly Journal of Experimental Psychology, 49B*, 60-80.

Domjan, M. (1980). Effects of the intertrial Interval on taste-aversion learning in rats. *Physiology & Behavior, 25*(1), 117-125.

Douglas, R. J., & Martin, K. A. C. (1991). Opening the grey box. *Trends In Neuroscience, 14*, 286-293.

Duch, W., & Jankowski, N. (1999). Survey of neural transfer functions. *Neural Computing Surveys, 2*, 163-212.

Durlach, P. J., & Rescorla, R. A. (1980). Potentiation rather than overshadowing in flavor-aversion learning: An analysis in terms of within-compound associations. *Journal of Experimental Psychology: Animal Behavior Processes, 6*, 175-187.

Egger, M.D., & Miller, N.E. (1963). When is a reward reinforcing: An experimental study of information hypothesis. *Journal of Comparative and Physiological Psychology, 56*, 132-137.

Eich, J. M. (1982). A composite holographic associative recall model. *Psychological Review, 89*, 627-661.

Enquist, M., & Ghirlanda, S. (2005). *Neural Networks and Animal Behavior*. Princeton: Princeton University Press.

Estes, W. K. (1975). Some targets for mathematical psychology. *Journal of Mathematical Psychology, 12*, 263-282.

Fodor, J. A. (1975). *The Language Of Thought*. Cambridge, MA: Harvard University Press.

Fodor, J. A., & Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture. *Cognition, 28*, 3-71.

Freeman, J. A. (1994). *Simulating Neural Networks With Mathematica*. Reading, MA: Addison-Wesley.

Freeman, J. H., Spencer, C. O., Skelton, R. W., & Stanton, M. E. (1993). Ontogeny of eye blink conditioning in the rat - Effects of US intensity and interstimulus-interval on delay conditioning. *Psychobiology, 21*(3), 233-242.

French, R. M. (1999). Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences, 3*(4), 128-135.

Gallistel, C. R. (1990). *The Organization Of Learning*. Cambridge, MA: MIT Press.

Gallistel, C. R., & Gibbon, J. (2000). Time, rate, and conditioning. *Psychological Review, 107*(2), 289-344.

Gerstner, W., & Kistler, W. M. (2002). Mathematical formulations of Hebbian learning. *Biological Cybernetics, 87*(5-6), 404-415.

Gibbon, J. (1977). Scalar expectancy theory and Weber's law in animal timing. *Psychological Review, 84,* 279-325.

Girden, E. (1940). Cerebral mechanisms in conditioning under curare. *American Journal of Psychology, 53*, 397-406.

Gluck, M. A. (1991). Stimulus-generalization and representation in adaptive network models of category learning. *Psychological Science, 2*(1), 50-55.

Gluck, M. A., & Bower, G. H. (1988). From conditioning to category learning - an adaptive network model. *Journal of Experimental Psychology-General, 117*(3), 227-247.

Gluck, M. A., & Myers, C. (2001). *Gateway to memory : an introduction to neural network modeling of the hippocampus and learning*. Cambridge, Mass.: MIT Press.

Grahame, N. J., Barnet, R. C., & Miller, R. R. (1992). Pavlovian conditioning in multiple contexts: Competition between contexts for comparator status. *Animal Learning & Behavior, 20*, 329-338.

Grey Walter, W. (1963). *The Living Brain*. New York: W.W. Norton & Co.

Grossberg, S. (1980). How does the brain build a cognitive code? Psychological review. *87, 1-51*.

Grossberg, S. (1988). *Neural Networks And Natural Intelligence*. Cambridge, MA: MIT Press.

Hammond, L. J. (1968). Retardation of fear acquisition by a previously inhibitory CS. *Journal of Comparative and Physiological Psychology, 66*, 756-759.

Harlow, H.F., & Stagner, R. (1933). Effect of complete striate muscle paralysis upon the learning process. *Journal of Experimental Psychology, 16*, 283-294.

Hebb, D. O. (1949). *The Organization Of Behaviour*. New York: John Wiley and Sons.

Hearst, E., & Franklin, S. R. (1977). Positive and negative relations between a signal and food: Approach-withdrawal behavior to signal. *Journal of Experimental Psychology: Animal Behavior Processes*, *3*, 37-52.

Hetherington, P., & Seidenberg, M. S. (1989). Is there “catastrophic interference” in connectionist networks? In *Proceedings of the 1989 meeting of the Cognitive Science Society* (pp. 26-33). Hillsdale NJ: Lawrence Erlbaum Associates.

Hinton, G. E., & Anderson, J. A. (1981). *Parallel Models Of Associative Memory*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Hsu, F., Anantharaman, T., Campbell, M., & Nowatzyk, A. (1990). A grandmaster chess machine. *Scientific American, 263*, 44-50.

James, W. (1890). *The Principles Of Psychology, Volume One*. New York, NY: Dover Publications.

Johnson-Laird, P. N. (1983). *Mental models*. Cambridge, MA: Harvard University Press.

Jordan, M. I. (1986). An introduction to linear algebra in parallel distributed processing. In D. Rumelhart & J. McClelland (Eds.), *Parallel Distributed Processing, Volume 1* (pp. 365-422). Cambridge, MA: MIT Press.

Kamin, L. J. (1968). Attention-like processes in classical conditioning. In M. R. Jones (Ed.), *Miami symposium on the prediction of behavior: Aversive stimulation* (pp. 9-32). Miami: University of Miami Press.

Kamin, L. J. (1969). Selective association and conditioning. In N. J. Mackintosh & W. K. Honig (Eds.), *Fundamental Issues In Associative Learning* (pp. 42-64). Halifax: Dalhousie University Press.

Kamin, L. J., & Schaub, R. E. (1963). Effects of conditioned stimulus intensity on conditioned emotional response. *Journal of Comparative and Physiological Psychology, 56*(3), 502-507.

Kaufman, M. A., & Bolles, R. C. (1981). A nonassociative aspect of overshadowing. *Bulletin of the Psychonomic Society, 18*, 318-320.

Kasabov, N. K. (1996). *Foundations Of Neural Networks, Fuzzy Systems, And Knowledge Engineering*. Cambridge, MA: MIT Press.

Kehoe, E. J. (1988). A layered network model of associative learning: Learning to learn and configuration. *Psychological review, 95*, 411-433.

Khallad, Y., & Moore, J. (1996). Blocking, unblocking, and overexpectation in autoshaping with pigeons. *Journal of the Experimental Analysis of Behavior, 65*(3), 575-591.

Kirkpatrick, K. (2002). Packet theory of conditioning and timing. *Behavioural Processes, 57*(2-3), 89-106.

Köhler, W. (1947). *Gestalt Psychology, An Introduction To New Concepts In Modern Psychology*. New York,: Liveright Pub. Corp.

Kohonen, T. (1977). *Associative Memory: A System-Theoretical Approach*. New York: Springer-Verlag.

Kohonen, T. (1984). *Self-organization and associative memory*. New York: Springer-Verlag.

Kovach, L. D. (1964). *Computer-Oriented Mathematics*. San Francisco, CA: Holden-Day.

Kraemer, P. J., Lariviere, N. A., & Spear, N. E. (1988). Expression of a Taste-Aversion Conditioned with an Odor Taste Compound - Overshadowing is relatively weak in weanlings and decreases over a retention interval in adults. *Animal Learning & Behavior, 16*, 164-168.

Kremer, E. F. (1978). The Rescorla-Wagner model: Losses in associative strength in compound conditioned stimuli. *Journal of Experimental Psychology: Animal Behavior Processes, 4*, 22-36.

Kruschke, J. K. (1992). ALCOVE: An exemplar-based connectionist model of category learning. *Psychological Review, 99*, 22-44.

Kruschke, J. K. (1996a). Base rates in category learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 22*, 3-26.

Kruschke, J. K. (1996b). Dimensional relevance shifts in category learning. *Connection Science, 8*, 225-247.

Krushke, J. K. (2001). Toward a unified model of attention in associative learning. *Journal of Mathematical Psychology, 45*, 812-863.

Krushke, J. K. (2003). Attention in learning. *Current Directions In Psychological Science, 12*, 171-175.

Kukla, A. (1989). Nonempirical issues in psychology. *American Psychologist, 44*(5), 785-794.

Lachter, J., & Bever, T. G. (1988). The relation between linguistic structure and associative theories of language learning - A constructive critique of some connectionist learning models. *Cognition, 28*, 195-247.

Lattal, K. M., & Nakajima, S. (1998). Overexpectation in appetitive Pavlovian and instrumental conditioning. *Animal Learning & Behavior, 26*(3), 351-360.

Lederman, L. (1993). *The God Particle*. New York, NY: Dell Publishing.

Lee, T. T. Y., Charrier, I., Bloomfield, L. L., Weisman, R. G., & Sturdy, C. B. (2006). Frequency-range discriminations and absolute pitch in black-capped chickadees (*Poecile atricapillus*), mountain chickadees (*Poecile gambeli*), and zebra finches (*Taeniopygia guttata*). *Journal of Comparative Psychology*, *120*, 217-228.

Levitan, I. B., & Kaczmarek, L. K. (1991). *The Neuron: Cell And Molecular Biology*. New York: Oxford University Press.

Lewandowsky, S. (1993). The rewards and hazards of computer simulations. *Psychological science, 4*, 236-243.

Lippmann, R. P. (1989). Pattern classification using neural networks. *IEEE Communications magazine, November*, 47-64.

Locke, J. (1977). *An Essay Concerning Human Understanding*. London: J.M. Dent & Sons.

Lubow, R. E. (2005). Construct validity of the animal latent inhibition model of selective attention deficits in schizophrenia. *Schizophrenia Bulletin, 31*(1), 139-153.

Lubow, R. E., & Moore, A. U. (1959). Latent inhibition - the effect of nonreinforced pre-exposure to the conditional stimulus. *Journal of Comparative and Physiological Psychology, 52*(4), 415-419.

Luce, R. D. (1997). Several unresolved conceptual problems of mathematical psychology. *Journal of Mathematical Psychology, 41*, 79-87.

Luce, R. D. (1999). Where is mathematical modeling in psychology headed? *Theory & Psychology, 9*, 723-737.

Lunneborg, C. E. (1994). *Modeling Experimental And Observational Data*. Belmont, CA: Duxbury Press.

Mackintosh, N. J. (1975). A theory of attention: Variation in the associability of stimuli with reinforcement. *Psychological Review, 82*, 276-298.

Marlin, N. A. (1982). Within-compound associations between the context and the conditioned stimulus. *Learning and Motivation, 13*, 526-541.

Marr, D. (1982). *Vision*. San Francisco, Ca.: W.H. Freeman.

Martinez, J. L., & Derrick, B. E. (1996). Long-term potentiation and learning. *Annual Review of Psychology, 47*, 173-203.

Martinez, J. L., & Kesner, R. (1998). *Neurobiology Of Learning And Memory*. San Diego, CA: Academic Press.

Matzel, L. D., Schachtman, T. R., & Miller, R. R. (1985). Recovery of an overshadowed association achieved by extinction of the overshadowing stimulus. *Learning and Motivation, 16*, 398-412.

Matzel, L. D., Shuster, K., & Miller, R. R. (1987). Covariation in conditioned response strength between elements trained in compound. *Animal Learning & Behavior, 15*, 439-447.

Mazur, J.E., Wagner A.R. (1982). An episodic model of associative learning. In Commons, M., Herrnstein, R., Wagner, A.R. (Eds.), *Quantitative Analyses of Behavior: Acquisition, Vol. 3*. (pp. 3-39). Cambridge, MA: Bellinger.

McClelland, J. L., & Rumelhart, D. E. (1986). *Parallel Distributed Processing, V.2*. Cambridge, MA: MIT Press.

McClelland, J. L., Rumelhart, D. E., & Hinton, G. E. (1986). The appeal of parallel distributed processing. In D. Rumelhart & J. McClelland (Eds.), *Parallel Distributed Processing* (Vol. 1, pp. 3-44). Cambridge, MA: MIT Press.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics, 5*, 115-133.

Medler, D. A. (1998). A brief history of connectionism. *Neural Computing Surveys, 1*, 18-72.

Miller, R. R. (2006). Challenges facing contemporary associative approaches to acquired behavior. *Comparative Cognition & Behavior Reviews, 1*, 77-93.

Miller, R. R., Barnet, R. C., & Grahame, N. J. (1995). Assessment of the Rescorla-Wagner model. *Psychological Bulletin, 117*, 363-386.

Miller, R.R., & Matzel, L.D. (1988). The comparator hypothesis: A response rule for the expression of associations. In G.H. Bower (Ed.), *The Psychology of Learning and Motivation*, *22*, 51-92.

Miller, R. R., & Spear, N. E. (1985). *Information Processing In Animals: Conditioned Inhibition*. Hillsdale, N.J.: L. Erlbaum Associates.

Milner, P. M. (1957). The cell assembly: Mark II. *Psychological Review, 64*, 242-252.

Minsky, M. (1985). *The Society Of Mind*. New York: Simon & Schuster.

Minsky, M., & Papert, S. (1988). *Perceptrons, 3rd Edition*. Cambridge, MA: MIT Press.

Mirman, D., & Spivey, M. (2001). Retroactive interference in neural networks and in humans: the effect of pattern-based learning. *Connection Science, 13*, 257-275.

Murdock, B. B. (1982). A theory for the storage and retrieval of item and associative information. *Psychological Review, 89*, 609-626.

Murdock, B. B. (1997). Context and mediators in a theory of distributed associative memory (TODAM2). *Psychological Review, 104*, 839-862.

Napier, R. M., Macrae, M., & Kehoe, E. J. (1992). Rapid reacquisition in conditioning of the rabbit's nictitating-membrane response. *Journal of Experimental Psychology: Animal Behavior Processes, 18*, 182-192.

Nickerson, C., Bloomfield, L. L., Dawson, M. R. W., & Sturdy, C. B. (2006). Artificial neural networks that discriminate notes from the chick-a-dee call of *Poecile atricapillus*: The effect of pitch transformations. *Journal of the Acoustical Society of America, 120*, 1111-1117.

Ortony, A. (1979). *Metaphor And Thought*. Cambridge: Cambridge University Press.

Pao, Y.-H. (1989). *Adaptive Pattern Recognition And Neural Networks*. Reading, MA: Addison-Wesley.

Parisi, D., Cecconi, F., & Nolfi, S. (1990). Econets: Neural networks that learn in an environment. *Network, 1*, 149-168.

Pavlov, I. P. (1927). *Conditioned Reflexes*. New York, NY: Oxford University Press.

Pearce, J. M. (1987). A Model for Stimulus-Generalization in Pavlovian Conditioning. *Psychological Review, 94*(1), 61-73.

Pearce, J. M. (1994). Similarity and discrimination: A selective review and a connectionist model. *Psychological Review, 101*, 587-607.

Pearce, J. M. (1997). *Animal Learning And Cognition: An Introduction*. East Sussex: Psychology Press.

Pearce, J. M. (2002). Evaluation and development of a connectionist theory of configural learning. *Animal Learning & Behavior, 30*, 73-95.

Pearce, J. M., & Bouton, M. E. (2001). Theories of associative learning in animals. *Annual Review of Psychology, 52*, 111-139.

Pearce, J. M., & Hall, G. (1980). A model for Pavlovian learning - Variations in the effectiveness of conditioned but not of unconditioned stimuli. *Psychological Review, 87*, 532-552.

Pfeifer, R., & Scheier, C. (1999). *Understanding Intelligence*. Cambridge, MA: MIT Press.

Pinker, S., & Prince, A. (1988). On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. *Cognition, 28*, 73-193.

Pitt, M. A., Myung, I. J., & Zhang, S. B. (2002). Toward a method of selecting among computational models of cognition. *Psychological Review, 109*, 472-491.

Polanyi, M. (1966). *The Tacit Dimension* ([1st ed.). Garden City, N.Y.,: Doubleday.

Pylyshyn, Z. W. (1984). *Computation And Cognition*. Cambridge, MA.: MIT Press.

Quinlan, P. (1991). *Connectionism And Psychology*. Chicago, IL: University of Chicago Press.

Rescorla, R. A. (1968). Probability of shock in presence and absence of Cs in fear conditioning. *Journal of Comparative and Physiological Psychology, 66*, 1-5.

Rescorla, R. A. (1969). Conditioned inhibition of fear resulting from negative CS-US contingencies. *Journal of Comparative and Physiological Psychology, 67*, 504-509.

Rescorla, R. A. (1970). Reduction in effectiveness of reinforcement after prior excitatory conditioning. *Learning and Motivation, 1*(4), 372-381.

Rescorla, R. A. (1971). Summation and retardation tests of latent inhibition. *Journal of Comparative and Physiological Psychology, 75*, 77-81

Rescorla, R. A. (1973). Evidence for the "unique stimulus" account of configural conditioning. *Journal of Comparative and Physiological Psychology, 85*, 331-338.

Rescorla, R. A. (1976). Stimulus generalization - Some predictions from a model of Pavlovian conditioning. *Journal of Experimental Psychology-Animal Behavior Processes, 2*(1), 88-96.

Rescorla, R. A. (1988). Pavlovian Conditioning - Its Not What You Think It Is. *American Psychologist, 43*(3), 151-160.

Rescorla, R. A. (1999). Summation and overexpectation with qualitatively different outcomes. *Animal Learning & Behavior, 27*(1), 50-62.

Rescorla, R. A. (2003). Contemporary study of Pavlovian conditioning. *The Spanish Journal of Psychology, 6*, 185-195.

Rescorla, R. A., & Wagner, A. R. (1972). A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. In A. H. Black & W. F. Prokasy (Eds.), *Classical Conditioning II: Current Research And Theory* (pp. 64-99). New York, NY: Appleton-Century-Crofts.

Ripley, B. D. (1996). *Pattern Recognition And Neural Networks*. Cambridge, UK: Cambridge University Press.

Rizzuto, D. S., & Kahana, M. J. (2001). An autoassociative neural network model of paired-associate learning. *Neural Computation, 13*(9), 2075-2092.

Robins, A. (1995). Catastrophic forgetting, rehearsal, and pseudorehearsal. *Connection Science, 7*(2), 123-146.

Rochester, N., Holland, J. H., Haibt, L. H., & Duda, W. L. (1956). Tests on a cell assembly theory of the action of the brain, using a large digital computer. *IRE Transactions On Information Theory, IT-2*, 80-93.

Roediger, H. L. (1990). Implicit memory - Retention without remembering. *American Psychologist, 45*(9), 1043-1056.

Roitblat, H. L., & von Fersen, L. (1992). Comparative cognition - Representations and processes in learning and memory. *Annual Review of Psychology, 43*, 671-710.

Rojas, R. (1996). *Neural Networks: A Systematic Exploration*. Berlin: Springer.

Rosenblatt, F. (1962). *Principles Of Neurodynamics*. Washington: Spartan Books.

Royce, J. R. (1970). The present situation in theoretical psychology. In J. R. Royce (Ed.), *Toward Unification In Psychology*. Toronto: University of Toronto Press.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature, 323*, 533-536.

Rumelhart, D. E., & McClelland, J. L. (1986). *Parallel Distributed Processing, V.1*. Cambridge, MA: MIT Press.

Savastano, H. I., Arcediano, F., Stout, S. C., & Miller, R. R. (2003). Interaction between preexposure and overshadowing: Further analysis of the extended comparator hypothesis. *Quarterly Journal of Experimental Psychology Section B: Comparative and Physiological Psychology, 56*, 371-395.

Schmajuk, N. A. (1997). *Animal Learning and Cognition : A Neural Network Approach*. Cambridge [England] ; New York, NY, USA: Cambridge University Press.

Schmajuk, N. A., & Blair, H. T. (1993). Place learning and the dynamics of spatial navigation: A neural network approach. *Adaptive Behavior, 1*, 353-384.

Schmajuk, N. A., & DiCarlo, J. J. (1992). Stimulus configuration, classical-conditioning, and hippocampal function. *Psychological Review, 99*(2), 268-305.

Schmajuk, N. A., Lam, Y. W., & Gray, J. A. (1996). Latent inhibition: A neural network approach. *Journal of Experimental Psychology: Animal Behavior Processes, 22*(3), 321-349.

Schmajuk, N. A., Lamoureux, J. A., & Holland, P. C. (1998). Occasion setting: A neural network approach. *Psychological Review, 105*, 3-32.

Schmajuk, N. A., & Larrauri, J. A. (2006). Experimental challenges to theories of classical conditioning: Application of an attentional model of storage and retrieval. *Journal of Experimental Psychology: Animal Behavior Processes, 32*(1), 1-20.

Schmajuk, N. A., Larrauri, J. A., & Labar, K. S. (2007). Reinstatement of conditioned fear and the hippocampus: An attentional-associative model. *Behavioural Brain Research, 177*, 242-253.

Shanks, D. R. (1995). *The Psychology Of Associative Learning*. Cambridge, UK: Cambridge University Press.

Siegelmann, H. T. (1999). *Neural Networks And Analog Computation: Beyond The Turing Limit*. Boston, MA: Birkhauser.

Simon, H. A., & Schaeffer, J. (1992). The game of chess. In R. J. Aumann & S. Hart (Eds.), *Handbook of Game Theory with Economic Applications* (Vol. 1, pp. 1-17): Elsevier Science Publishers, Netherlands.

Smolensky, P. (1988). On the proper treatment of connectionism. *Behavioral and Brain Sciences, 11*, 1-74.

Solomon, P.R., Lohr, C.A., & Moore, J.W. (1974) Latent inhibition of the rabbit's nictitating response: summation tests for active inhibition as a function of a number of CS preexposures. *Bulletin of the Psychonomic Society, 4*, 557-559.

Solomon, R. L., & Turner, L. H. (1962). Discriminative classical conditioning in dogs paralyzed by curare can later control discriminative avoidance responses in the normal state. *Psychological Review, 69*, 202-219.

Stone, G.O. (1986). An analysis of the delta rule and the learning of statistical associations. In Rumelhart, D.E. & McClelland, J.L., eds. Parallel Distributed Processing, Vol. 1. Cambridge, MA: MIT Press (pp. 444-459).

Sutton, R. S., & Barto, A. G. (1981). Toward a modern theory of adaptive networks: Expectation and prediction. *Psychological Review, 88*(2), 135-170.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement Learning*. Cambridge, MA: MIT Press.

Thorndike, E. L. (1932). *The Fundamentals Of Learning*. New York, NY: Bureau Of Publications, Teachers College, Columbia University.

Van Hamme, L. J., & Wasserman, E. A. (1994). Cue competition in causality judgments: The role of nonpresentation of compound stimulus elements. *Learning and Motivation, 25*, 127-151.

Varela, F.J., Thompson, E., & Rosch, E. (1991). *The Embodied Mind*. Cambridge, MA: MIT Press.

Vogel, E.H., Brandon, S.E., & Wagner, A.R. (2003). Stimulus representation in SOP: II. An application to inhibition of delay. *Behavioural Processes, 62*, 27-48.

Wagner, A.R. (1981). SOP: a model of automatic memory processing in animal behavior. In Spear, N.E., Miller, R.R. (Eds.) Information Processing in Animals: Memory Mechanisms. (pp. 5-47) Hillsdale, NJ: Erlbaum, Hillsdale.

Wagner, A. R. (2003). Context-sensitive elemental theory. *Quarterly Journal of Experimental Psychology Section B-Comparative and Physiological Psychology, 56*(1), 7-29.

Walkenbach, J., & Haddad, N. F. (1980). The Rescorla-Wagner theory of conditioning - a review of the literature. *Psychological Record, 30*(4), 497-509.

Warren, H. C. (1921). *A History Of The Association Psychology*. New York, NY: Charles Scribner's Sons.

Wasserman, E. A., & Miller, R. R. (1997). What's elementary about associative learning? *Annual Review of Psychology, 48*, 573-607.

Webb, B., & Consi, T.R. (2001). *Biorobotics: Methods and Applications*. Menlo Park, CA: AAAI Press/MIT Press.

Weisman, R., Brownlie, L., Olthof, A., Njegovan, M., Sturdy, C., & Mewhort, D. (1999). Timing and classifying brief acoustic stimuli by songbirds and humans. *Journal of Experimental Psychology: Animal Behavior Processes, 25*, 139-152.

Wickens, T. D. (1982). *Models For Behavior: Stochastic Processes In Psychology*. San Francisco, CA: W.H. Freeman.

Widrow, B. (1962). Generalization and information storage in networks of ADALINE "neurons". In M. C. Yovits, G. T. Jacobi & G. D. Goldsteing (Eds.), *Self-Organizing Systems 1962* (pp. 435-461). Washington, DC: Spartan Books.

Widrow, B., & Hoff, M. E. (1960). Adaptive switching circuits. *Institute of Radio Engineers, Western Electronic Show and Convention, Convention Record, Part 4*, 96-104.

Wiener, N. (1948). *Cybernetics, or Control and Communication in the Animal and the Machine*. Cambridge, MA: MIT Press.

Williams, B. A., & McDevitt, M. A. (2002). Inhibition and superconditioning. *Psychological Science, 13*(5), 454-459.

Wilson, R.A. (2004). *Boundaries of the Mind: The Individual in the Fragile Sciences: Cognition*. New York: Cambridge University Press.

Yaremchuk, V., Willson, L. R., Spetch, M. L., & Dawson, M. R. W. (2005). The implications of null patterns and output unit activation functions on simulation studies of learning: A case study of patterning. *Learning and Motivation, 36*(1), 88-103.

Zimmer-Hart, C. L., & Rescorla, R. A. (1974). Extinction of Pavlovian conditioned inhibition. *Journal of Comparative and Physiological Psychology, 86*, 837-845.

