1992

A simplified drive-reinforcement model for unsupervised learning in artificial neural networks

David B. Suits

Rochester Institute of Technology
Computer Science Department

A Simplified Drive-Reinforcement Model for Unsupervised Learning in Artificial Neural Networks

by

David B. Suits

A thesis, submitted to
The Faculty of the Computer Science Department
in partial fulfillment of the requirements for the degree of
Master of Science in Computer Science.

Approved by:

Professor Peter G. Anderson, R.I.T.

Professor John A. Biles, R.I.T.

Dr. Arun Rao, Eastman Kodak Company

31 Aug 92

Title of Thesis: A Simplified Drive-Reinforcement Model for Unsupervised Learning in Artificial Neural Networks

I hereby grant permission to the Wallace Memorial Library, of RIT, to reproduce my thesis in whole or in part. Any reproduction will not be for commerical use or profit.

David B. Suits

1 Sept 1992
Date

ABSTRACT

Partly in response to the apparent limitations of explicit symbol processing used by traditional artificial intelligence research, there has been, within the last decade, a growing interest in artificial neural networks. This thesis focuses on the development and testing of a model for describing certain kinds of biological phenomena.

The many artificial neural networks available may be classified into three types: (1) self-organizing networks, which have input but no feedback; (2) unsupervised networks, requiring minimal feedback (perhaps a signal indicating success or failure); and (3) supervised models, which employ far more extensive (and, I think, biologically implausible) feedback mechanisms. In this thesis I examine only models of the second type.

The Rescorla-Wagner "trial-level" model gives a quantitative description of what happens as a result of a conditioning trial. But that model, along with more detailed, "temporal" (i.e., intratrial) models, such as a traditional Hebbian model and the Sutton-Barto model, make predictions which are at odds with empirical data. Klopf's "drive-reinforcement" model is a much more robust account, from which I develop a simplified drive-reinforcement (SDR) model. I prepare a number of experiments to test my SDR model's correspondence with empirical data derived from animal learning experiments; I demonstrate that the model is capable of describing a wide variety of classical conditioning phenomena; and I show how the model may form the basis for instrumental conditioning as well. Finally, I add a simple motivating principle (or "drive") and show that such an addition seems to enhance the learning capabilities of the model.

CONTENTS

5. Instrumental Conditioning

6. Directions for Future Research

1. Introduction: GOFAI vs. ANNs

1.1 GOFAI and the rationalist tradition

In 1976 Newell and Simon proposed their Physical Symbol System Hypothesis: "A physical symbol system has the necessary and sufficient means for general intelligent action", where by "general intelligence" they meant intelligence of the same scope as human intelligence (Newell and Simon, 1976). This is perhaps one of the clearest statements of the presumption held by many modern researchers in the field of artificial intelligence (or what Haugeland, 1985, called "Good Old Fashioned Artificial Intelligence"). The presumption is that certain kinds of manipulation of explicit information are the sine qua non for intelligence (or at least for construing behavior as intelligent). Long before 1976, however, the search had been on for the proper kinds of explicit representations and the proper kinds of manipulation of that information. (One can trace this back at least as far as Leibniz in the 17th century.) What has emerged from this premise has been a flurry of successful, initial, exciting starts in various areas of research, followed by sad realizations that what is needed to extend those successes is computational power of such magnitude that it is, well, mind-boggling. This, the symbol manipulation approach to simulations (or synthesizations) of intelligence, now seems best suited for severely restricted domains (e.g., expert systems).

Forceful explanations of the poverty of explicit symbol manipulation have been advanced. (For examples, Dreyfus, 1979; Dreyfus and Dreyfus, 1988; Searle,

1980, Johnson, 1987.) Apparently, it is the explicitness of both the symbols and their manipulations which is a central problem. Is it possible, then, to devise artificial systems which harbor, in some fashion, implicit knowledge representations and manipulation, and which, in some fashion, synthesize intelligent behavior from non-intelligent parts?

One such approach involves the modeling of small biological units, under the assumption that simple units, linked together in massively parallel architectures, might be able to accomplish at least some of the tasks which the symbol manipulation approach had the most difficulty with, especially all those ambiguous tasks which fall under the heading of pattern recognition.

1.2 Artificial neural networks

The biological unit under investigation here is the neuron, or else a network of neurons. Since the investigation of real biological neurons is still fraught with puzzles and uncertainties, we will discuss "artificial neurons", whose properties are very much simplified from what is known and conjectured about biological neurons.

A simple artificial neuron is a device which computes an output as some function of its inputs (figure 1.1). The following are some considerations, following the biological metaphor, which might be useful for specifying the nature of such neurons.

We will want to know whether there should be upper or lower bounds on the number of inputs to a neuron. We will also need to specify whether these inputs are to be inhibitory or excitatory. Similarly for the neuron's output: although a neuron will have only one output, it is allowed to branch off to become the input to an arbitrary number of other neurons. How will a neuron be designated

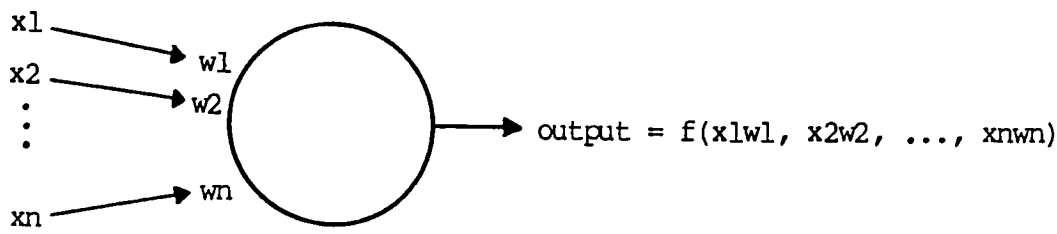


Figure 1.1. An artificial neuron computes its output as some function of its weighted inputs.

as excitatory or inhibitory? That is, will a distinction be made between kinds of neurons, or will some of a neuron's outputs be designated as excitatory and others inhibitory? There is neurophysiological evidence that some neurons are excitatory to other neurons a short distance away, but inhibitory for neurons much futher away.

The inputs to an artificial neuron might be modified by some value (constant or variable) called an input's weight or efficacy. There is some neurophysiological parallel to this in the sometimes changing synapse strengths of neural connections; some synapses can even be permanently changed.

An artificial neuron, like its biological counterpart, might not be allowed to "fire" until the input signal strength (perhaps the sum of its excitations less the sum of its inhibitions) reaches or passes some threshold. We will want to know what this threshold is, whether it should be the same for all neurons, and whether it is ever allowed to change (and under what conditions).

After "firing", a neuron might be required to undergo a "recharging" -- refractory — period, during which its output would be inactive. When designing an artificial neuron, we must decide what this refractory period is, whether it is to be the same for all neurons, and whether it is allowed to change.

If a neuron receives input during its refractory period, or if the input has not yet passed the neuron's threshold value, what happens to the input? We may choose to maintain the input until the neuron is ready to deal with it, or we may have the input "decay" or weaken during that period.

We may wish to specify some upper bound on the strength of the input, beyond which the neuron will be said to be saturated. Supersaturating a neuron might be one way to cause it to become inhibited or to shut down temporarily with "fatigue".

What will be the outputs of neurons? Will they be limited to binary values (+1 and 0, or +1 and -1)? Biological neurons generally distinguish their input by the frequency of the input firings. In artificial neurons, we might represent such frequencies by means of a range of values (say, real numbers between 0 and 1).

Not only is there a wide selection of possible neuron types, there is a practically infinite variety of ways of connecting neurons together, from fully connected nets to randomly connected nets, and from single-layered to multi-layered. Nets may be designed so as to eventually reach a stable state, or so as to oscillate among certain states, or neither.

Although the main tradition in artificial neural network models assumes that learning is a process of synaptic weight changing, it must be kept in mind that there are other processes which may take place instead of, or in addition to, weight changes. For examples, new neurons may grow and old ones die, and synaptic connections may be made and lost. In the mammalian brain there is initially a superabundance of neurons; many more exist early on than survive to maturity. There is, similarly, an initial overproduction of axons, and environmental conditions may well determine which connections are maintained. (Black &

Greenough, 1986; Crutcher, 1986; Lynch, 1987.)

I propose to classify artificial neural networks into three broad classifications: (1) Type 1 are networks which have input, but no external feedback and no error correction. They are quite simply self-organizing. von der Malsburg's (1973) suggestion for self-organization in the visual cortex is a good example. So are Kohonen (1984) and Hopfield (1982). (2) Type 2 networks have feedback, but of a minimal sort — perhaps a single signal indicating failure (Barto, Sutton & Anderson, 1983) or else success (Klopf, 1986; Sutton & Barto, 1981; the SDR model presented below in Chapter 4 is also such a model). (3) Type 3 have more explicit feedback representing the full response which the net ought to have produced, or the difference between the net's actual output and the expected output. (So-called "back-propagation" has become such a popular, off-the-shelf learning algorithm. See Rumelhart, Hinton & Williams, 1986.)

Learning algorithms with a "teacher" -- a feedback mechanism which has perfect knowledge of the appropriate response -- are called supervised. (Anderson, 1988, p. 124.) Type 3 networks, then, would certainly be supervised, and type 1 certainly would be unsupervised. But what of type 2? In most examples of type 2, a feedback signal would not be an indication of "perfect knowledge" on the part of the feedback mechanism, even though the signal is informative. I shall therefore call examples of type 2 unsupervised. And it is with a certain kind of unsupervised learning that this thesis will be concerned.

Why focus on unsupervised methods? My principal motive is to find some connection with biological systems. There is evidence of both types of unsupervised learning in animals (I shall be concentrating on classical and instrumental conditioning), but the connection between biological systems and an all-knowing "teacher" is very tenuous: supervised learning is possible only when a

correct response is known and can be given to the net, and many lived situations make this either impractical or impossible — or inappropriate. Consider the problem of simulating a rat in a training area; a piece of cheese is to its west. (Figure 1.2) To simplify the issue, let there be constraints on the rat's movements: north, east, south and west only. The question now is, what is the proper move? That is, what is the proper motor output which the rat is to be taught?

One is sorely tempted to answer, "Move west." But why? The answer no doubt presumed a more or less precise problem which the rat is to solve (whether the rat knows it or not), namely, How to get to the food? There was no doubt another presumption: cheese is food. And another: the rat is hungry. And still another: the rat is to move in the most efficient (time/space-wise) direction to the food.

But there is no reason a priori why the rat must move west if it is not hungry, if cheese is not agreeable to its palate, or if efficiency is not

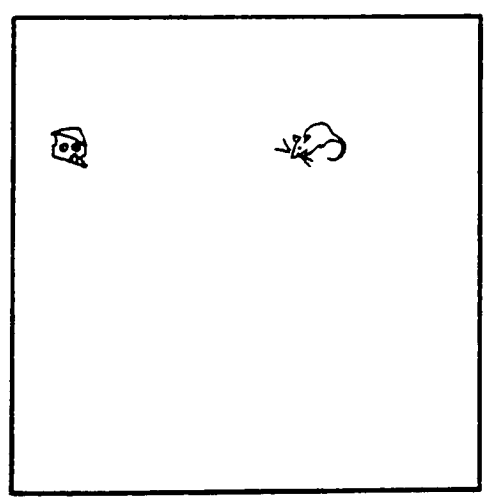


Figure 1.2. Which way should the rat move?

pleasing. That is, it is we, the creators of the rat and its world, who set the problem and its parameters, and specify the answer, rather than letting the rat discover, for itself, what pleases and displeases it, what kinds of actions are satisfying and what kinds are not. If the rat is hungry, it might prefer moving expeditiously toward food. Unless it is more tired than hungry, or more afraid than hungry, or .... Surely a rat learns even in the absence of omniscient feedback.

In Chapters 5 and 6 below, I shall return to this issue of a hypothetical rat learning its way towards its goals on the basis of minimal feedback. But first a foundation in classical conditioning must be laid.

1.3 The plan of this thesis

In this thesis, I will investigate a learning routine for a single artificial element. Such an element might be a single neuron or it might be a small net of neurons. For the sake of simplicity, I will generally speak of such an element as though it were a single neuron. The learning routine is inspired by animal learning experiments called conditioning (classical and instrumental). How close to the experimental data can a simple, artificial element be made to perform? And is there reason to believe that such a simple element might form the basis for both classical and instrumental conditioning?

Chapter 2 will examine classical conditioning and discuss the notion of a stimulus trace as an aid in explaining conditioning phenomena.

Chapter 3 will present the Rescorla-Wagner insight which generated a quantitative description of some basic facts of classical conditioning. But a more detailed treatment will be needed in order to be consistent with empirical data, and so a traditional Hebbian model and the Sutton-Barto model will be examined. Finally, Klopf's drive-reinforcement model will be presented as the most com-

plete model.

In Chapter 4 I will develop a simplified drive reinforcement (SDR) model as a variation of Klopf's. I will present a number of experiments to show that the SDR model is consistent with many — but not all — classical conditioning phenomena.

In Chapter 5 I will extend the SDR model's application to include simple instrumental conditioning experiments. In addition, I will suggest an enhancement of the model by means of a simulated "drive" which affects synaptic weights. I will present several experiments to show that such an addition can speed the learning process.

Finally, Chapter 6 will tie up some loose ends and mention some possibilities for further experimentation with the SDR model.

2. Classical Conditioning

2.1 Basic Phenomena

2.1.1 First order conditioning

Classical conditioning (also called Pavlovian conditioning) involves a learned relationship or association among a number of stimuli presented to an organism and a particular behavioral response of the organism.

In a simple learning experiment, one stimulus will initially produce no observable behavioral response, while a different stimulus will reliably evoke a response. Under certain circumstances, the first, neutral stimulus can eventually be made to evoke the same, or nearly the same, response as the second stimulus, in which case the originally neutral stimulus is said to have been conditioned.

The stimulus which initially produced no response is called the conditioned stimulus (CS), and the stimulus which reliably produced a response is called the unconditioned stimulus (US, or UCS). The animal's response — if any — to the CS is called the conditioned response (CR), and the response to the US is the unconditioned response (UR, or UCR). Conditioning, therefore, involves creating a CR; and often — but not always — the CR will resemble the UR.

Pavlov's (1927) investigations produced not only a wide variety of data concerning conditioning, but also some speculations on underlying mechanisms to account for the phenomena. The most salient data concern certain features of the

CR as a function of the temporal relationship between the CS and the US.

Consider one of Pavlov's famous experiments: A dog is found to salivate reliably in the presence of food, but makes no observable response to the presentation of a tone (perhaps from a bell). But if the tone is paired with the food — such that over a number of trials the tone slightly precedes the presentation of food — then the dog will begin more and more reliably to salivate in response to the tone even in the absence of the food. The dog has been conditioned to "anticipate" or "expect" food on the basis of the tone. Or perhaps we may say that the CS (the tone, in this case) comes to reliably "predict" the US (the food). Salivation in response to the US is the UR. The CR is at first nothing; during conditioning the CR comes more and more to resemble the UR, i.e., the dog begins to salivate at the tone. The acquisition of a CR often describes a S-shaped curve as depicted in figure 2.1.

2.1.2 Extinction, reacquisition and spontaneous recovery

If, after acquisition, the animal is presented with the CS which is not followed by the US, then, after a number of such trials, the CR will begin to

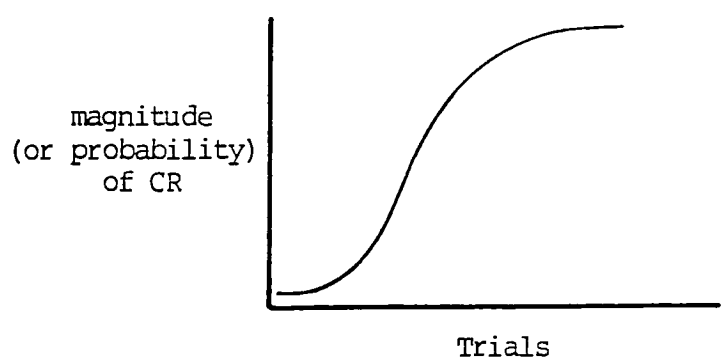


Figure 2.1. The acquisition of a conditioned response (CR) over time usually describes an S-shaped curve. (After Schwartz and Lacey, 1982, p. 47.)

fade. In cognitive terms, the CS loses its former predictive ability. The CR may be revived, however, by once again pairing the CS with the US. This is called reacquisition. In general, reacquisition occurs faster than acquisition. Often, an extinguished response will spontaneously recover after a rest period. Indeed, "almost all investigators since Pavlov have found spontaneous recovery of some degree following extinction" (Young, 1961, p. 367).

2.1.3 Second (and higher) order conditioning

Once a CR has been acquired, it may under certain conditions act as the US for an even earlier CS. Suppose, for example, a dog has been conditioned to associate a tone with food. If the tone is from then on preceded by, say, a light, even without the subsequent presentation of food, then the dog might come to salivate in response to the light alone. Under rare circumstances a third CS, coming before the light, might be made to evoke the CR. However, higher order conditioning is generally short-lived and unreliable, since it involves the use of a previously conditioned CS (now playing the role of the US), which, not being followed by the original US, is itself undergoing extinction.

2.1.4 Blocking, compound conditioning, and overshadowing

Imagine two CSs, CS1 and CS2 (probably of two different sense modalities). After CS1 has been conditioned, then CS2, presented simultaneously with CS1, will usually not undergo conditioning. In cognitive terms, we might say that CS2 offers no information not already provided by CS1. The prior conditioning of CS1 is said to block conditioning of CS2.

If, however, both CS1 and CS2 are initially presented together as a compound, then both individually will be capable of producing the CR, although each separately will have less effect than the compound.

Finally, if, during acquisition using a compound of CS1 and CS2, one stimulus is markedly stronger (more "salient") than the other (perhaps, for example, CS1 is a loud tone, and CS2 is a dim light), then the stronger stimulus will overshadow the weaker. After acquisition using the compound, the animal will respond fairly reliably to the loud tone alone, but not reliably — or perhaps not at all — to the dim light alone.

2.1.5 Temporal relations between CS and US

Delay conditioning is an experimental procedure wherein the US onset comes after the CS onset but at or before the CS offset. If US onset comes after CS offset, it is called trace conditioning. (Figure 2.2.) Delay conditioning is, in general, more efficacious than trace conditioning. According to Kamin (1965),

> a few hundred msec. seem very critical when interposed between CS termination and US [onset], although a full minute interposed between CS onset and US [onset, in delay conditioning] produces no detectable decremental effect (p. 141),

and when

> CS onset is "favorably close" to the US ..., there is no significant difference between delayed and trace procedures. When ... CS onset is "too far" from the US, excellent conditioning occurs with a delayed procedure, but virtually none with a trace procedure (p. 142).

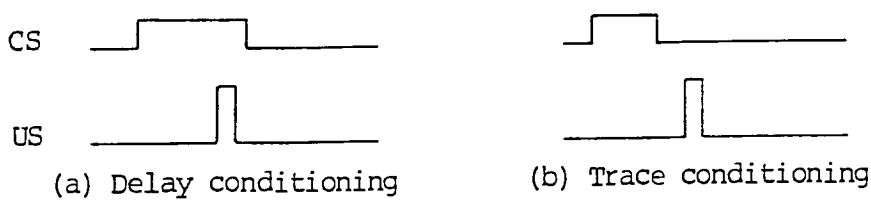


Figure 2.2. Timing diagrams for delay conditioning and trace conditioning.

Pavlov's (1927) experiments in delay conditioning revealed an optimum interstimulus interval (ISI) — the time between CS onset and US onset — of approximately .5 second. But conditioning is obtained with much longer ISIs in both delay and trace conditioning (Rachlin, 1976; Bitterman, 1965; Ost & Lauer, 1965; Razran, 1965; Gormezano, 1972; Alkon, 1983; Alkon, et. al., 1989; Kehoe, 1990). The efficacy of delay conditioning as a function of the ISI is sketched in figure 2.3. In general, for each preparation there is some optimum ISI where conditioning is most effective, and efficacy drops off as the ISI is less than or greater than that optimum. Little or no conditioning occurs at ISIs of zero.

There have been some tentative suggestions that conditioning might occur with negative ISIs. This is called backward conditioning because the US comes before the CS. But it is generally agreed that ISIs less than some small positive value provide little if any conditioning (Kamin, 1965).

2.1.6 Temporal features of the CR generation

In early stages of conditioning, the CR tends to appear temporally near the US; but as conditioning proceeds, the CR tends to appear earlier — there is a foreshortening of CR onset latency. There have also been some attempts to measure a minimal CR onset latency. (Kehoe, 1990.) For longer ISIs (greater than 200 msec), there is some evidence (Pavlov, 1927) that not only will the CR not

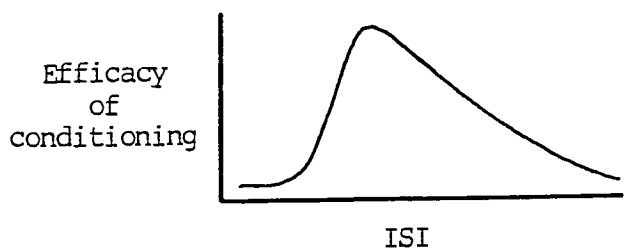


Figure 2.3. The optimum ISI (Interstimulus Interval) is often, but not always, abont 450 - 500 msec. (After Gormezano, 1972, p. 163.)

appear during the early portions of the ISI, but the early portions of the CS may actually be inhibitory. (Pavlov also reported that CR onset latency can be suddenly changed by the introduction of a novel stimulus during the ISI.)

As a CR is acquired, the amplitude of the CR (for example, nictitating membrane extension in the rabbit eye nictitating membrane experiments) tends to peak at some point near the US onset. If the ISI is changed, the peak of the CR will change with it. And if a CR is acquired with a random mix of two different ISIs, the CR will appear with two peaks, reflecting the two ISIs. (Kehoe, 1990.)

Experiments have been performed wherein one group of animals was originally conditioned for a short time with an ISI of 250 msec; other groups were conditioned with ISIs ranging from 500 msec to 4000 msec. After 15 trials, none produced a CR, but each group was then further conditioned at a common ISI of 500 msec. The 250 msec group was from then on more easily conditioned; the rate of acquisition dropped off smoothly from 500 to 4000 msec. (Kehoe, 1990.)

When only one conditioning trial per day is used, the optimal ISI appears closer to 1000 msec, versus 250 msec when 5 to 100 trials per session are used.

> For all ISI values, a reduction in the number of trials per session produces a dramatic increase in the rate of CR acquisition. For example, in a one trial-per-day procedure using 400-msec ISI, only 30 CS-US pairings are needed for a group of subjects to achieve a mean level of 90 percent CRs. When 50 trials are presented per day using the 400-msec ISI and a 60-sec ITI [inter trial interval], more than 200 pairings are required to achieve 90 percent CRs. (Kehoe, 1990, p. 399.)

Still other features of conditioning will be of concern to us, and they will be examined in Chapter 4 when the SDR (Simplified Drive-Reinforcement) Model is presented.

2.2 The Stimulus Trace

Conditioning does not take place unless the US occurs after the CS. But how could a later reinforcement (the US) strengthen the effect of an earlier stimulus (the CS)? Lest there be a temptation to postulate some sort of "backwards causation", it should be remembered that it is not the CS's effects which are strengthened, but rather the effects of future CSs. On the theory here adopted, a reinforcing US strengthens a synaptic weight, and the effect of such a weight change is not revealed until the next CS occurs.

There is, nevertheless, a related problem: Since the CS and US are temporally separated, how does conditioning occur across such a "temporal gap"? The problem is not unlike the problem of action at a distance: How does one thing here manage to affect another thing over there? The nearly universal distrust for anything resembling action at a distance prompts various theories of intermediaries. So, too, in the case of CS-US association. Pavlov (1927) invented the concept of a "stimulus trace", championed also by Hull (1943), although what that amounted to in neurological terms was not clear. Mowrer (1973) suggested that such a stimulus trace might have been envisioned as a kind of "reverberation" in the nervous system which exists for a short while after neuronal excitation, fading eventually to zero (figure 2.4). This might account for why a response will be affected more at the beginning of the trace than later in the trace, but it otherwise does not provide sufficient detail to count as a satisfactory mechanism for conditioning. In addition, such a trace is simply a continuation of input (and/or output) activity. We will see later, however, that it is advantageous to follow Klopf's (1979) suggestion that the stimulus trace be something separate from neuronal activity, such that a stimulus trace is itself non-stimulating.

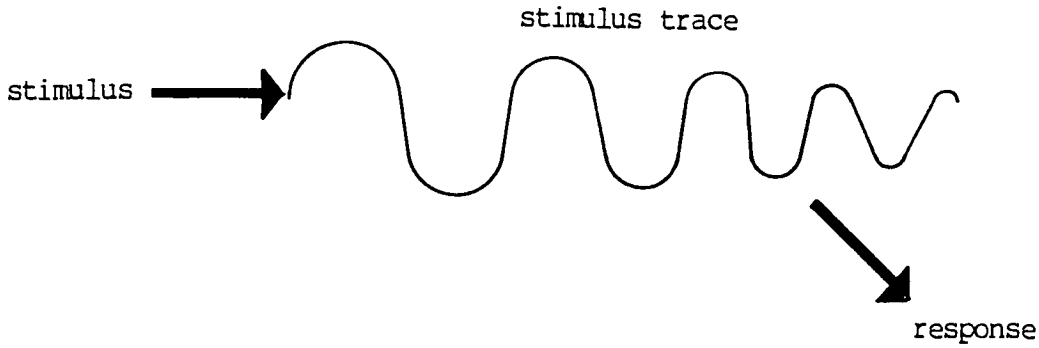


Figure 2.4. An early conception of a stimulus trace suggests that it might be some sort of reverberation caused by the original stimulus. (After Mowrer, 1973, p. 362.)

However the stimulus trace is defined, it performs two major functions. First, it influences the rate of associative learning at a given interstimulus interval. A US which arrives earlier or later than an optimal time will have less effect on learning than a US which arrives at some optimal time. Second, the stimulus trace provides the "driving force" for the CR.

2.3 Three words of caution

It is apparently not simply temporal contiguity between the CS and the US which is important in Pavlovian conditioning (and, as we shall see later on, in instrumental conditioning as well). As Schwartz and Lacey (1982) say,

> Animals seem to evaluate the <u>information</u> conveyed by various stimuli, and selectively to associate the most informative stimuli with USs. For conditioning to occur, a potential CS must tell the animal something about the US which it would not otherwise know. (p. 49)

So temporal contiguity and a differential <u>contingency</u> between the CS and the US are necessary. That's the first word of caution.

The second has to do with the nature of the conditioned stimulus. Arbitrary CSs paired with arbitrary USs will not, in general, produce conditioning. In one

experiment, conditioning failed in rats when a light was used as the CS but succeeded when a buzzer was used. (Young, 1965.) And a rat may be conditioned to certain tastes of food (CS) if paired with later sickness (US), whereas a light or a noise accompanying the food will not condition to later sickness (Young, 1965; Rachlin, 1976). One simply has to try a stimulus and see if it works. (On the other hand, usually a stimulus which acts as a reinforcer for conditioning one kind of behavior in a given organism will also act as a reinforcer for other behaviors of that same organism.)

The third word of caution has to do with cognitive language. Terms such as "information" and "tell the animal something", which Schwartz and Lacey use, or "anticipate", "expect" and "predict", which I used earlier, ought to be kept in scare quotes because they tend to imply cognitive activities separate from the conditioning itself, as though the animal might be an observer of the CS-US contingency, and, by a process of rational induction, come to have certain beliefs about that contingency. But that there are such separate cognitive functions is not guaranteed by the behavioral data alone; even very primitive animal organisms are conditionable. (I am not aware of any conditioning experiments on plants.) It was Pavlov's assumption that conditioning is necessary to explain intelligent behavior:

> Are there any grounds for differentiation, for distinguishing between that which the physiologist calls the temporary connection and that which the psychologist terms association? They are fully identical; they merge and absorb each other. (Pavlov, 1955, p. 251.)

> It cannot be doubted that the fundamental laws governing the activity of the first signal system [i.e., the conditioned reflex mechanisms which humans share with all other animals] must also govern that of the second [i.e., speech], because it, too, is activity of that same nervous tissue. (Pavlov, 1955, p. 262.)

And Skinner (1938), to whose ideas we shall return in Chapter 5, reminded us often that there is no warrant for insisting that intelligence is anything other than intelligent behavior.

3. Models of Classical Conditioning

3.1 Kinds of models

Gluck, Reifsnider & Thompson (1990) divide models of classical conditioning into three general types:

(1) Trial-level models account for the net effects of a given training trial on the CS-US association. Such models are silent on the exact temporal relationship between the stimuli, and have no way of accounting for the real-time properties of CR generation. The Rescorla-Wagner model (discussed below) is perhaps the most well-known example of a trial-level model.

(2) Temporal models account for ISI effects and temporal primacy effects such as second and higher order conditioning. But as with trial-level models, temporal models do not seek to describe the moment-by-moment effects of CR generation. Examples of temporal models include Donegan & Wagner (1987), Mazur and Wagner (1982), Wagner (1981), Sutton & Barto (1981, 1987), Klopf (1988), and the SDR model described in the next chapter. See also Gluck, Parker & Reifsnider (1988, 1989).

(3) Finally, **real-time models** attempt to describe the precise topology of CR generation. Examples are Desmond (1990), Moore & Blazis (1989) and Gluck, Reifsnider & Thompson (1990).

This is not to say, however, that trial-level models cannot be enhanced so as to account for temporal or real-time phenomena. Gluck, Reifsnider & Thompson (1990), for example, attempt to extend the Rescorla-Wagner trial-level model so

as to describe some real-time features of CR generation.

The SDR model presented below in Chapter 4 is based upon Klopf's "differential Hebbian" model (which he also calls a "drive-reinforcement" model, and, using a classification scheme somewhat different from the one used here, categorizes his model as a "real-time" model), and is a temporal model only: it does not aspire to describe the real-time details of CR generation (about which more later), but it does account for many temporal primacy effects which trial-level models leave out.

It is not the intention of this thesis to investigate all models of classical conditioning, but rather to discuss briefly those few models which provide background for the SDR model. Grossberg, for example, has done extensive work on modelling both classical and instrumental conditioning, along with the phenomena of attention and motivation. (Grossberg, 1971a, 1971b, 1974, 1982, 1987, 1990; Grossberg & Levine, 1987; see also Buonomano, Baxter & Byrne, 1990 and Buonomano & Byrne, 1990.) But his relatively complex models are considerably different from the approaches discussed in this thesis. (The two neural net models which form the basis for my SDR model are criticized by Grossberg, 1990.)

3.2 The Rescorla-Wagner model

In their trial-level model, Rescorla and Wagner (1972) accounted for various phenomena of compound conditioning (such as blocking and overshadowing) by suggesting a quantification of the CR and UR, along with the assumption that the asymptote of the CR learning curve is a function of the UR. According to their theory, the amount of conditioning <u>on a given trial</u> for a single CS is:

$$\Delta CR = \beta(UR - CR) \tag{3.1}$$

where

ΔCR = the change in CR strength,

UR = the strength of the UR,

CR = the strength of the CR before the trial, and

β = a learning rate constant, $0 < \beta < 1$.

That is, on each trial, conditioning changes by some fraction of the amount of conditioning which the UR will support. This results in the negatively accelerated learning curve of figure 3.1a. During extinction, when US is absent, UR = 0, and so ΔCR is negative. Consequently, on each extinction trial, the CR decreases by some fraction of the CR still remaining. (Figure 3.1b.)

To apply the equation to compound conditioning with two CSs, Rescorla and Wagner assumed that the total CR strength is the sum of the strengths of its elements:

$$CR_{total} = CR1 + CR2 \tag{3.2}$$

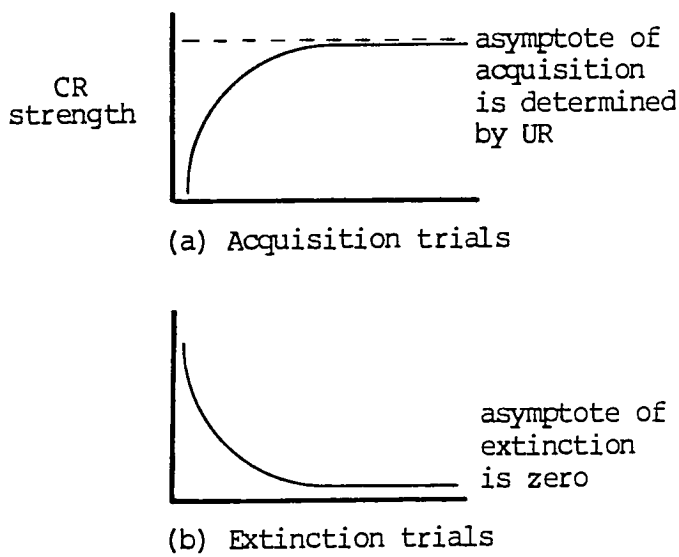


Figure 3.1. Graphs of the predictions made by the Rescorla-Wagner model of classical conditioning. (After Rachlin, 1976.)

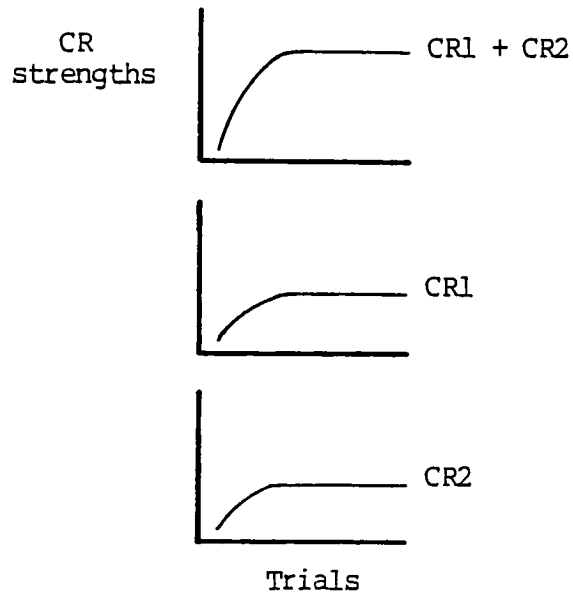


Figure 3.2. The Rescorla-Wagner model predictions for two CSs (here assumed to be of equal salience). (After Rachlin, 1976.)

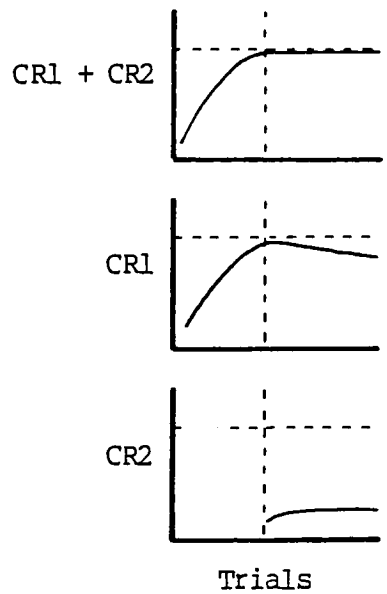


Figure 3.3. The relative strengths of two CSs during a blocking experiment.

(see figure 3.2). Thus, equation 3.1 is a special case of equation 3.2, namely, when one of the CSs is absent.

Equation 3.3 will account for the phenomena of blocking and overshadowing. In a blocking experiment, one stimulus, CS1, is paired with the US to produce conditioning; then trials of a compound, CS1 + CS2, are paired with the US. CS2 is found to gain strength only to the extent that conditioning to CS1 is less than the maximum determined by the UR (figure 3.3). Similarly, in an overshadowing experiment, where conditioning is to the compound CS1 + CS2, but where CS1 is stronger (more salient) than CS2, the total effect of the compound is divided unequally between the constituents. (Figure 3.4.)

Equation 3.2 also accounts for more complex compound conditioning phenomena. Consider an experiment by Wagner and Saavedra (Rescorla & Wagner, 1972), which I will repeat in Chapter 4 using the SDR model. Three groups of rabbits underwent slightly different conditioning regimens (on the eyelid nictitating membrane,

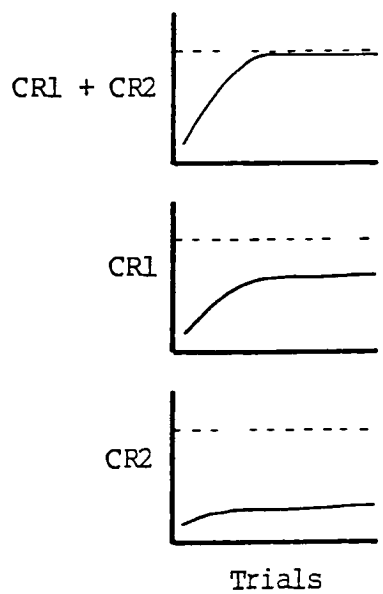


Figure 3.4. The relative strengths of two CSs during an overshadowing experiment.

wherein a CS such as a tone or a light is followed by a puff of air to the cornea — the US — which causes the membrane to close). On every other trial, each group received conditioning to the compound CS1 + CS2. (Assume equal salience of the two stimuli.) The three groups differed in what took place during the alternate trials. (See figure 3.5, top.) For group 1, CS2 was eliminated (i.e., conditioning continued with CS1 alone). For group 2, CS1, CS2 and US were absent (i.e., there was a rest period). And for group 3, CS1 was presented without the US (i.e., CS1 underwent an extinction trial). The results are

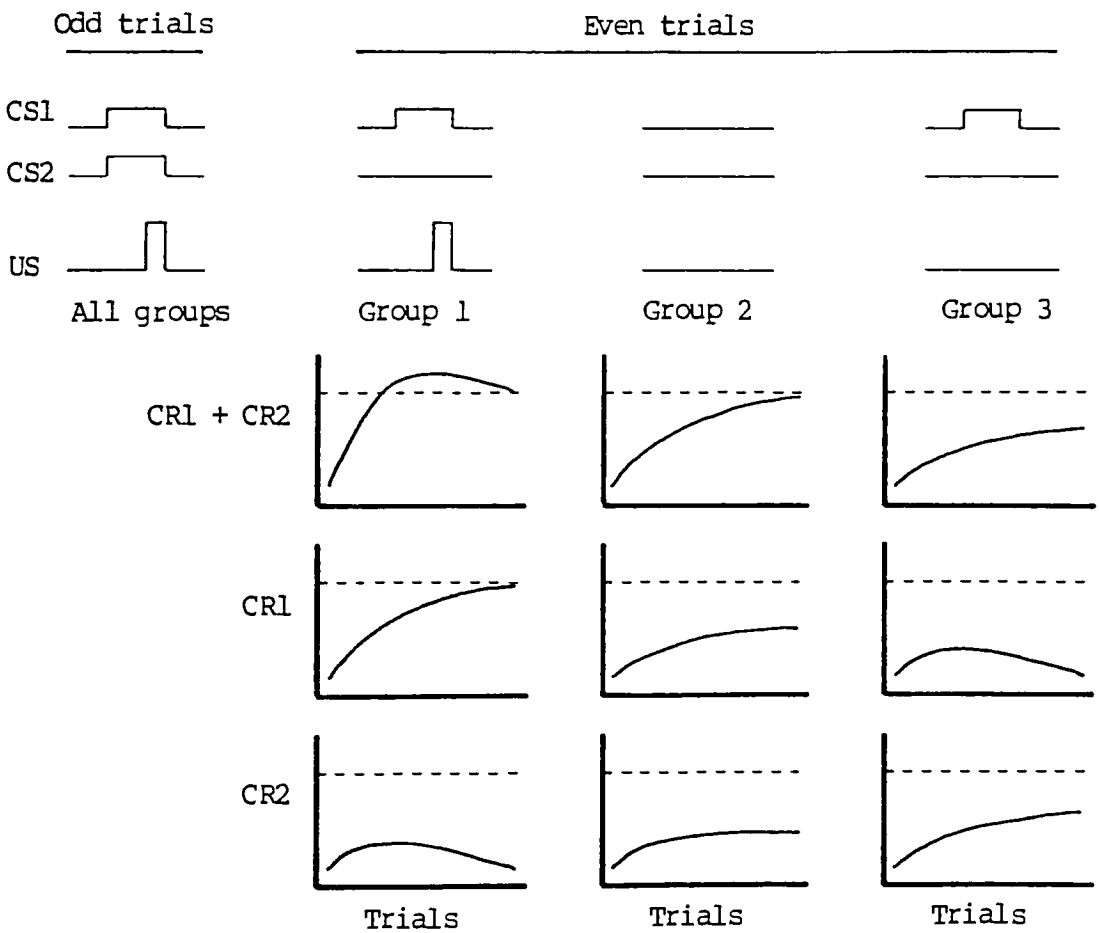


Figure 3.5. The Wagner-Saavedra experiment. Top: timing diagram. Bottom: Results for the CRs.

sketched in the bottom portion of figure 3.5. Notice that for group 1, the total response is for a short time pushed above the asymptote (determined by UR), because CR1 is always increasing, whereas CR2 is effective only every other trial, and on alternate trials remains the same. Since equation (2) requires that the total response be brought closer to the asymptote, CR1 + CR2 now begins to decline; since CR1 is always paired with US, CS2 now begins to have an inhibitory effect on every other trial. For group 2, the total response is constantly divided equally between CR1 and CR2. And for group 3, CS1 undergoes extinction every other trial, while CR2 increases on alternate trials. Notice that the three groups were treated indentically with respect to CS2, yet CS2 emerged with quite different properties (the bottom row of graphs in figure 3.5).

The Rescorla-Wagner model, however, suffers from several inaccuracies as a model of actual animal learning data. First, the model predicts the extinction of conditioned inhibition, yet such extinction in actual animal learning experiments turns out to be very difficult (Desmond, 1990; Miller & Spear, 1985). (Sutton and Barto, 1990, point out that the model's prediction of extinction of conditioned inhibition can be corrected if the sum of all CSs times their saliencies is interpreted such that it is always non-negative.) Second, the model predicts a negatively accelerated acquisition curve, whereas experimental data suggest that the curve is usually initially positively accelerating, as in figure 2.1. Third, the model is an intertrial level model only, and does not account for certain intratrial phenomena such as delay vs. trace conditioning effects, serial compound conditioning (wherein CSs are presented non-simultaneously), and real-time CR topography. Finally, the model does not account for the more rapid reacquisition of CRs after initial acquisition and subsequent

extinction; nor does it account for the small degree of spontaneous recovery of a CR after acquisition and subsequent extinction.

3.3 The Hebbian Model

Perhaps the simplest artificial neural net model of classical conditioning is derived from D. O. Hebb's (1949) suggestion that

> 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.

A similar idea was announced earlier by William James (1890): "When two elementary brain processes have been active together or in immediate succession, one of them on reoccurring tends to propagate its excitement into the other." Such an approach may be seen as an outgrowth of British Associationism going back at least to Hobbes and Locke in the 17th century, although they tended to talk in terms of ideas rather than in terms of neural mechanisms.

The usual interpretation of Hebb's suggestion is that synaptic connections are strengthened on the basis of purely local — and simultaneous — correlations between neural units, and that the rule for changing the efficacy of the connection (or, as we shall call it, the weight) of synapse i at time t is:

$$\Delta w_i(t) = \beta[x_i(t)\ y(t)] \tag{3.3}$$

where $\Delta w_i(t) = w_i(t+1) - w_i(t)$; $x_i(t)$ is the presynaptic activity of synapse i; $y(t)$ is the postsynaptic activity (i.e., node output), which is the sum of all the presynaptic activations times their respective weights (the w_i); and β is a positive constant which determines the rate of learning.

But notice that because the model correlates simultaneous input and output

signals, it cannot adequately account for temporal difference effects in learning. (Carew, et. al., 1984; Wigstom & Gustafsson, 1983.) Figure 3.6 shows the results of pairing six different CSs with a constant US. As expected, the simulations confirm that the model predicts no associative learning unless the CS and US overlap. Thus, trace conditioning effects are not accounted for. Moreover, the acquisition curves are nearly linear, whereas empirical results give an initially positively accelerating curve.

Of course, the quote above from Hebb need not be taken as requiring simultaneous correlation. (Kosko, 1986, says that the operative argument for using

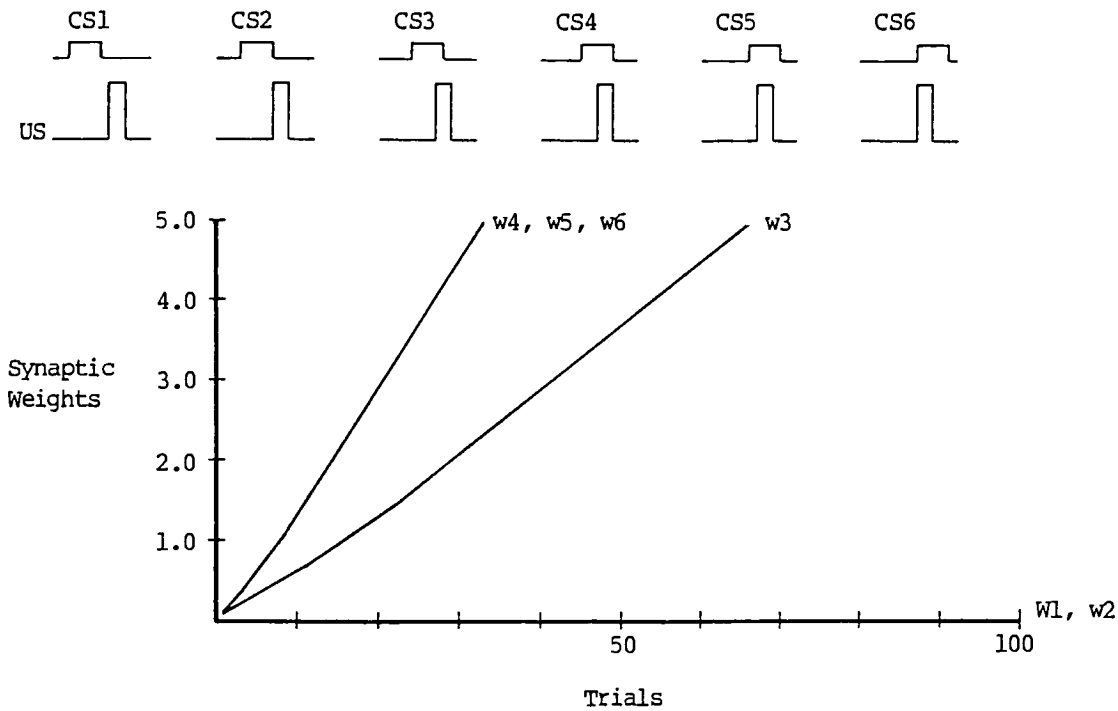


Figure 3.6. Timing diagram and synaptic weight curves for an artificial neuron following a Hebbian rule. Six different, independent tests were performed, representing six slightly different conditioning protocols: trace conditioning (CS1), four varieties of delay conditioning (CS2 through CS5), and simultaneous conditioning (CS6). The graph shows the Hebbian model predictions. CS4, CS5 and CS6 give identical results because these CSs are on during the whole of the US. CS3 overlaps only half the US, and it causes a slower weight increase. CS1 and CS2 are not simultaneous with any part of the US, and hence cause no weight increase. But empirical results indicate that CS1 through CS5 should produce different curves, with w1 the lowest and w5 the highest. CS6 should produce little or no weight change. Also, the Hebbian model incorrectly predicts approximately linear curves.

such an interpretations of Hebb "is simply that everyone uses it.")

A more fruitful assumption involves attention not to input and output signals, but rather to their changes. Any model which incorporates such an assumption we may call a differential Hebbian model. One such early model was developed by Sutton & Barto (1981), which is presented below. A. Harry Klopf, to whose suggestions (Klopf, 1979, 1982) Sutton & Barto are indebted, proposed a differential Hebbian model in 1986. (Independently, Kosko, 1986, proposed a similar model.) We will look at Klopf's model later, in preparation for the presentation of the SDR model, which is a simplified version of Klopf's.

3.4 The Sutton-Barto Model

In addition to an input, x (i.e., the CS), Sutton and Barto (1981, 1990) make use of a separate stimulus trace, $\bar{x}$, which is simply the weighted average of the values of x for some preceding time interval and does not itself produce any node output. They also postulate an output trace, $\bar{y}$, which is a weighted average of the output for some preceding time interval. As usual, the node output, y, is the sum of all presynaptic activity times their respective weights. $\bar{x}$ may be called an eligibility trace governed by

$$\bar{x}_i(t+1) = \alpha\bar{x}_i(t) + x_i(t) \tag{3.4}$$

and $\bar{y}$ may be called an expectation, described by

$$\bar{y}(t+1) = \gamma\bar{y}(t) + (1-\gamma)y(t) \tag{3.5}$$

where $0 \leq \alpha,\gamma < 1$. Synaptic efficacy (weight) is changed according to

$$\Delta w_i(t) = \beta[y(t) - \bar{y}(t)]\bar{x}_i(t) \tag{3.6}$$

where β is a positive learning rate constant. A weight is modified only if its

eligibility trace is non-zero and the current output, y, differs from the trace $\overline{y}$.

In the simplest (approximate) case, $\gamma = 0$, and so $\overline{y}(t) = y(t-1)$. Similarly, if $\alpha = 0$, then $\overline{x}_i(t) = x_i(t-1)$. And the rule for weight changes becomes

$$\Delta w_i(t) = \beta[y(t) - y(t-1)]x_i(t-1) \quad . \tag{3.7}$$

Whereas the Hebbian rule is usually taken to represent a correlation between input signals and output signals, the Sutton-Barto model attempts to capture the significance of changes in output, and consequently the model is more faithful to the temporal nature of conditioning. But there remain important deficiencies. Figure 3.7 repeats the conditions under which the Hebbian model was tested, but this time with the Sutton-Barto model. Several deficiencies in the model are readily apparent. A negatively accelerated acquisition curve is predicted by the model. Figure 3.7 also shows (as Klopf, 1986, 1988 points out) that the model incorrectly predicts conditioned inhibition when the CS and US substantially overlap (see Gormezano, 1972), because the trace $\overline{x}$ is continually increasing between CS onset and offset. Finally, there should be little or no conditioning when the CS and US onsets occur together, yet the Sutton-Barto model predicts a strong inhibition. Thus, the model will accurately predict empirical data only under quite restricted conditions. (Sutton and Barto, 1990, propose a "fix" for these problems in a substantial variation of their original model.)

Nevertheless the Sutton-Barto model represents a significant improvement on both the Rescorla-Wagner and the Hebbian models, because it accounts for some CR latency and ISI dependence. In addition, a Sutton-Barto adaptive element tends to find the earliest, most reliable predictors of the US, and tends to ignore any others. For example, in serial compound conditioning, even when both of two

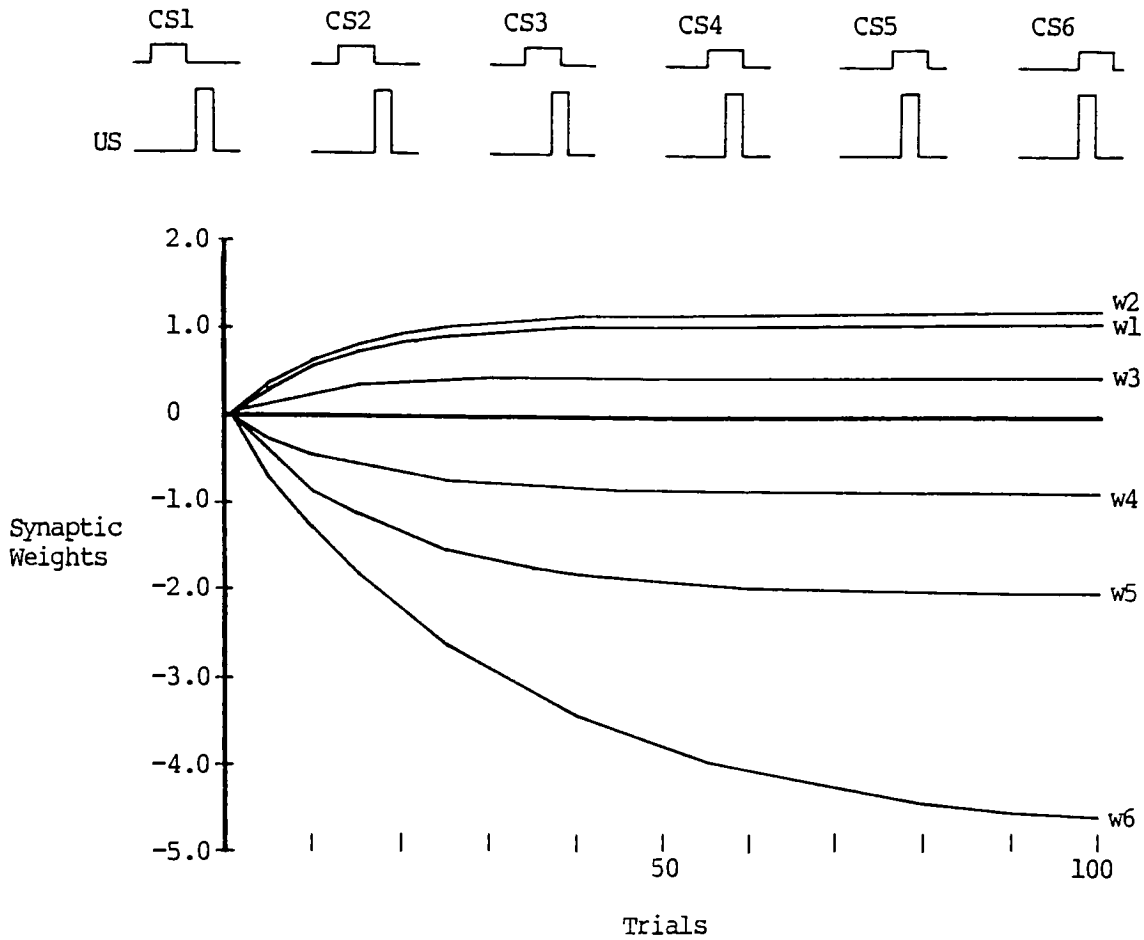


Figure 3.7. The Sutton-Barto model in six independent conditioning protocols. The model fails in cases where the CS and the US overlap: there should be no conditioned inhibition (w4, w5 and w6), and the asymptote for w3 should be higher than the asymptote for w2.

CSs are reliably correlated with the US, if one CS is earlier than the second, then — even if the second is initially dominant — the first will eventually dominate the second. (See also Kehoe & Schrueurs, 1987.) The Rescorla-Wagner model, being a trial-level model only, cannot account for such temporal effects.

3.5 Klopf's Drive-Reinforcement Model

Klopf (1982) proposed to treat neurons (or a class of neurons) as "hedonists". If depolarization (i.e., excitation) is considered "pleasure" and hyperpolarization (i.e., inhibition) as "pain", then we may imagine a neuron which

seeks to maximize depolarization and minimize hyperpolarization (or, rather, maximize their difference). Klopf calls the strategy which such a neuron might use heterostatic adaptation. It is quite straightforward: whenever the neuron fires (because its excitations less its inhibitions exceed some threshold), it will "notice", during some subsequent interval, τ (probably a few seconds), whether the difference between its excitations and inhibitions changes. Positive changes result in the neuron's greater tendency to fire on subsequent occasions, and negative changes have the opposite effect. Such tendencies will be implemented by means of changes to the neuron's postsynaptic efficacy — increasing the efficacy of excitatory connections in the case of positive changes, and increasing the efficacy of inhibitory connections in the case of negative changes.

> For a neuron, temporal and spatial configurations of active synapses represent conditioned stimuli (CS), firing represents a conditioned response (CR), and the excitation or inhibition that arrives during a limited period of time after firing constitutes the unconditioned stimulus (US) (Klopf, 1982, p. 5),

which then acts as CS for signals which arrive still later, etc.

Klopf's (1986, 1987) weight updating rule is:

$$\Delta w_i(t) = \Delta y(t) \sum_{j=1}^{\tau} c_j |w_i(t-j)| \min[0, \Delta x_i(t-j)] \qquad (3.8)$$

where $\Delta w_i(t) = w_i(t+1) - w_i(t)$; $\Delta y(t) = y(t) - y(t-1)$; $0 \leq y(t) \leq YMAX$; $y(t)$, the node output at time t, is the sum of all inputs, x_i, times their respective weights; $\Delta x_i(t) = x_i(t) - x_i(t-1)$; τ is the longest interstimulus interval over which conditioning is effective; and c_j is a learning rate constant which is proportional to the efficacy of conditioning when the interstimulus interval is

j. (That is, the constants c_1, c_2, ..., c_τ describe a stimulus trace — or eligibility — curve, with $c_1 > c_2 > \ldots > c_\tau$.) Weights have initial non-zero values and minimum absolute values — they do not cross zero. Weights are never allowed to reach zero for at that point no further weight changes could ever occur, because according to the weight updating rule a weight is a factor in subsequent weight changes. Klopf believes it to be biologically plausible that excitatory synapses remain excitatory and inhibitory synapses remain inhibitory.

Figure 3.8 illustrates the progress of a weight change using Klopf's updating rule during a typical delay conditioning trial. Notice that only changes

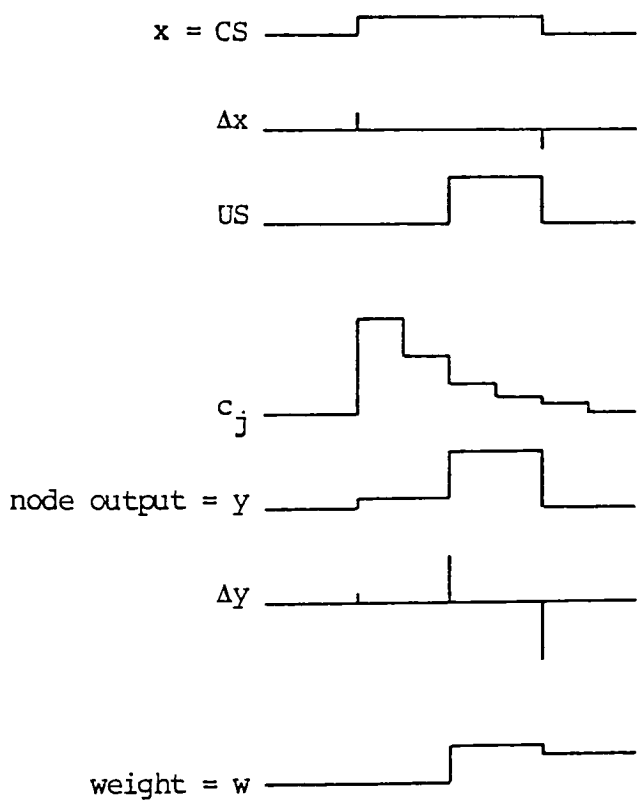


Figure 3.8. Klopf's Drive Reinforcement model during a single conditioning trial. The synaptic weight will change in the same direction as Y, but only when Y changes after a positive change in CS, with a smaller resulting weight change the more temporally distant the input and output changes are.

in neuronal output subsequent to CS activation will cause a weight change. Notice also that the weight initially increases on account of a positive change in neuronal output (at US onset), then falls slightly on subsequent negative output changes. But the decline is less than the rise because the negative output changes are temporally farther removed from the initial CS onset, and so the eligibility trace value (the c_j constant) is lower, resulting in less effect on the weight. ("With both delayed and trace procedures," says Kamin, 1965, p. 143, "the change in stimulation appears critical; and in each case an increase in stimulation produces more conditioning than does the corresponding reduction.")

An unusual coincidence of conditions could delight the perverse experimenter: Introduce a CS with any desired amplitude and weight; precisely at CS offset, introduce the US with amplitude equal to the CS times its weight. The result will be no change in the output until US offset. But that will be a negative change, and so the weight will decline. That is, the CS will effectively undergo an extinction trial. This will be repaired, however, when the neuron experiences a positive output change during the next trial at US onset, at which time the weight increases; the increase will more than make up for the previous slight decline, thus precluding further extinction — unless the perverse experimenter insists on adjusting the US amplitude downward an appropriate amount with each trial. We may reasonably presume that such precision is unlikely to be supplied in any actual learning experiment, and so the perverse experimenter will have to remain a Gedankenexperimenter only.

Klopf's model presents a slight advance over the Sutton-Barto model, in that it more accurately predicts learning rate curves (figure 3.9). Notice that the sixth curve — a case of "simultaneous conditioning" — is correctly predicted

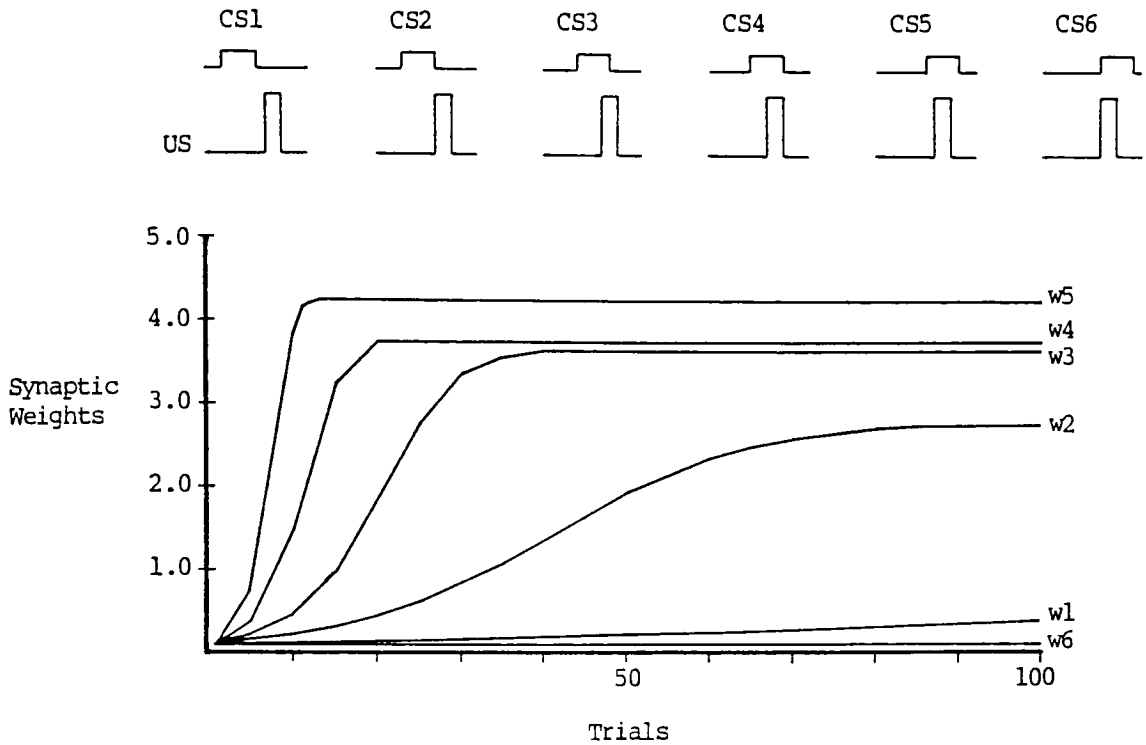


Figure 3.9. Klopf's Drive Reinforcement model under six independent conditioning protocols. The model's predictions accord well with empirical results, including the lack of conditioning when CS onset and US onset are simultaneous. The learning rate constants were 5.0, 3.0, 1.5, .75 and .25.

by the model to not participate in associative learning. But there are certain real-time features of conditioning which Klopf's model does not account for; these were mentioned earlier (in section 2.1.6) and will be discussed in the next chapter, with reference to my SDR model, which is based upon Klopf's, and which shares similar limitations.

4 The SDR Model

4.1 Background

On Klopf's drive-reinforcement model, earlier changes in synapse activation are to be correlated with later changes in neuronal activity. This is accomplished by discretizing a certain interval of time and assigning "eligibility" values to each discrete time — in effect, an array of values for the past τ time units. A second requirement is in effect a second array, namely, the record of changes of presynaptic activations, so that when neuronal activation changes, the past presynaptic activations can be correlated with the past eligibility values so as to create changes in synaptic weight.

But these two requirements impose a computational burden (especially in large networks), and so a simplifying alternative to one or both of them would be desirable. In addition, lists of eligibility values and presynaptic changes are necessarily finite, so that without changing the lists, we cannot take a more fine-grained look at weight changes, nor investigate conditioning effects for interstimulus intervals greater than the last discrete time unit, nor investigate conditioning effects for different eligibility values.

Klopf (1986) makes several not unreasonable simplifying stipulations: that the optimum ISI is 500 msecs; that ISIs shorter than the optimum are not interesting; and that ISIs greater than about 2500 msecs are not interesting. But these stipulations are very restricting. In particular, in animal learning experiments the optimum ISI varies depending on the experimental preparation and

the species of animal to be conditioned (Rachlin, 1976; Bitterman, 1965; Ost & Lauer, 1965; Razran, 1965; Gormezano, 1972; Alkon, 1983; Alkon, et.al., 1989), and conditioning effects occur well beyond 2.5 seconds (Ost & Lauer, 1965; Kehoe, 1990). In one experiment the ISI was 60 seconds. (Garcia, McGowan & Green, 1972.)

Since Klopf's eligibility values fall roughly on an exponential curve, my simplified drive reinforcement model uses a single value which acts as an exponentially decaying impression of past presynaptic changes, thereby both simplifying and generalizing Klopf's model. The exponential decay may be followed out to any arbitrarily large ISI; there is no requirement to keep lists of eligibility values and past presynaptic changes; and consequently the computational burden is considerably lessened.

4.2 The model

$$\Delta w_i(t) = \beta e_i(t) \Delta y(t) \tag{4.1}$$

$$e_i(t) = \alpha e_i(t-1) + |w_i(t-1)| \min[0, \Delta x_i(t-1)] \tag{4.2}$$

$$e_i(0) = 0 \tag{4.3}$$

where $\Delta w_i(t) = w_i(t+1) - w_i(t)$; $\Delta y(t) = y(t) - y(t-1)$; $y(t)$ is the node output, defined as the sum of all inputs, x_i, times their respective weights, w_i, and is bounded to $0 \leq y(t) \leq YMAX$ (probably any convenient function will do; I have used both hard-limiting and an exponential function); $\Delta x_i(t) = x_i(t) - x_i(t-1)$; and all weights, w_i, have initial non-zero values and minimum absolute values, $0 < WMIN \leq |w_i|$. β is a positive constant which influences the rate of weight changes. The e_i, always non-negative, are the "eligibility" values and act as exponentially decayed impressions of past positive presynaptic changes with decay rate α, $0 < \alpha < 1$.

The equations above can be described in terms of a simple algorithm for updating the synapses of a given node at time t:

$y = \text{bound}(\sum_i w_i x_i)$

$\Delta y = y - \text{previous\_}y$

for each synapse, i {

$e_i = e_i \alpha$

if $\Delta x_i > 0$ then $e_i = e_i + \Delta x_i |w_i|$

$w_i = w_i + \beta e_i \Delta y$

if $|w_i| < \text{WMIN}$ then $|w_i| = \text{WMIN}$

$\Delta x_i = x_i - \text{previous\_}x_i$

$\text{previous\_}x_i = x_i$

}

$\text{previous\_}y = y$

Although the SDR model was developed principally from Klopf's model, it may also be derived from the Sutton-Barto model in a straightforward way: Sutton-Barto's eligibility trace is generated directly from the level of presynaptic activity, x. But if instead the (positive) change in presynaptic activity times the synaptic weight is used, and if the neuronal output trace $\bar{y}$ is made as simple as possible by setting γ in equation 3.5 to 0, then the SDR model is the result. These changes are significant: the discordances between model predictions and empirical results vanish.

Figure 4.1 shows a time trace of the interrelationships of the main variables of the SDR model during some typical conditioning trials. Notice that the eligibility, e, is triggered by the rising edge of the CS (i.e., the node input x). Neither the falling edge of CS nor a steady level of CS (whether low or

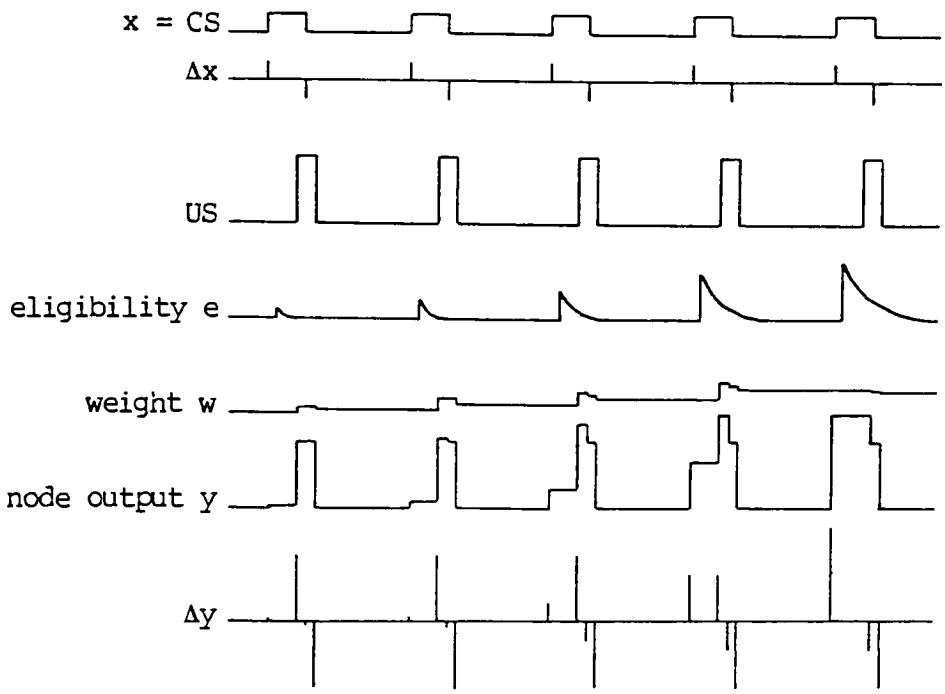


Figure 4.1. A time trace of the SDR model during several typical conditioning trials. Except for the eligibility values, the model acts approximately like Klopf's (see figure 3.8).

high) has any effect on e, although the node output y, and therefore also the change in ouput, are affected. Notice also that, following Klopf, the eligibility trace is simplified so that it has zero rise time, which is to say that when the difference between CS onset and US onset falls below some minimum, the CS and US are treated as being simultaneous. (For the purposes of microanalysis, the eligibility curve could be made more complex so that equation 4.2 would apply only beginning some finite but very small time after CS onset, before which some other appropriate equation would describe the shape of the (fast) rising eligibility trace.) The SDR model, following Klopf, requires that weights have minimum absolute values. Why should we begin with non-zero weights? For two reasons. First, the equations of the model will not produce weight changes when weights are zero. For this reason also, weight changes are not allowed to cross zero: excitatory weights remain excitatory, and inhibitory weights remain in-

hibitory. Second, a CR cannot be reinforced unless the CR has some possibility of occurring (prior to US onset).

4.3 Classical conditioning experiments with the SDR model

4.3.1 Trace conditioning, delay conditioning effects and extinction

Figure 4.2 shows a graph which repeats the tests given to the previous models. The results are from program SDR1.C (see Appendix A) over six separate runs, recording the progress of synaptic weight changes in an SDR unit over 100 trials using trace conditioning, delay conditioning and "simultaneous conditioning". For CS1, onset is at 2, offset at 6. Four separate versions of delay conditioning are presented, with four CSs, each four times units in duration,

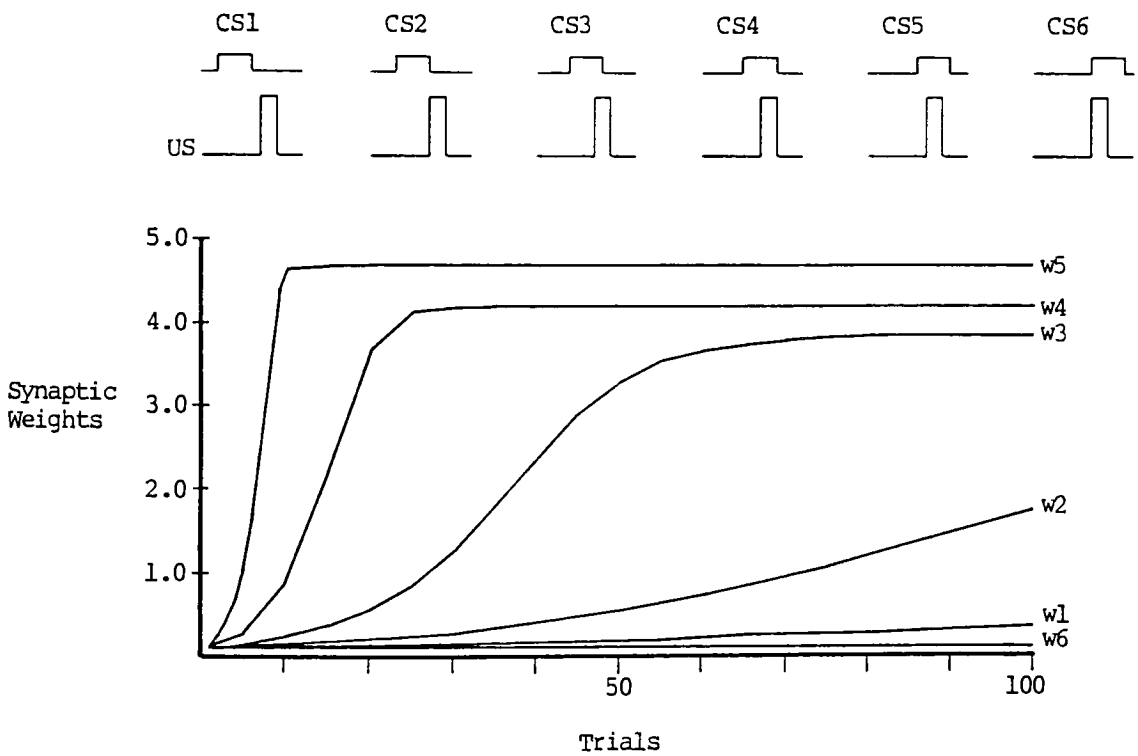


Figure 4.2. The SDR model under six conditioning experiments. (Compare with figures 3.6, 3.7 and 3.9.) Parameter values are given in the text.

but with shifted onsets: CS2 onset = 3, CS2 offset = 7; CS3 onset = 4, CS3 offset = 8; CS4 onset = 5, CS4 offset = 9; and CS5 onset = 6, CS5 offset = 10. In all cases, US onset is at 7 and US offset is at 9. US amplitude is 0.7 and amplitude of all CSs is 0.2. α, the decay rate constant, is 0.4, and β, the learning rate constant, is 1.5.

As expected, a synaptic weight (w1, w2, w3, w4 and w5 in figure 4.2) is shown to change more rapidly and approach a higher asymptote when its onset is closer to US onset. Each curve is S-shaped, corresponding to acquisition curves obtained in animal learning experiments. (See figure 2.1.) In program SDR1.C, node output is clipped at a maximum (1.0). Different node output functions may be employed (for example, y = 1 - exp(-y)) without disturbing the basic relationships of the synaptic weight curves.

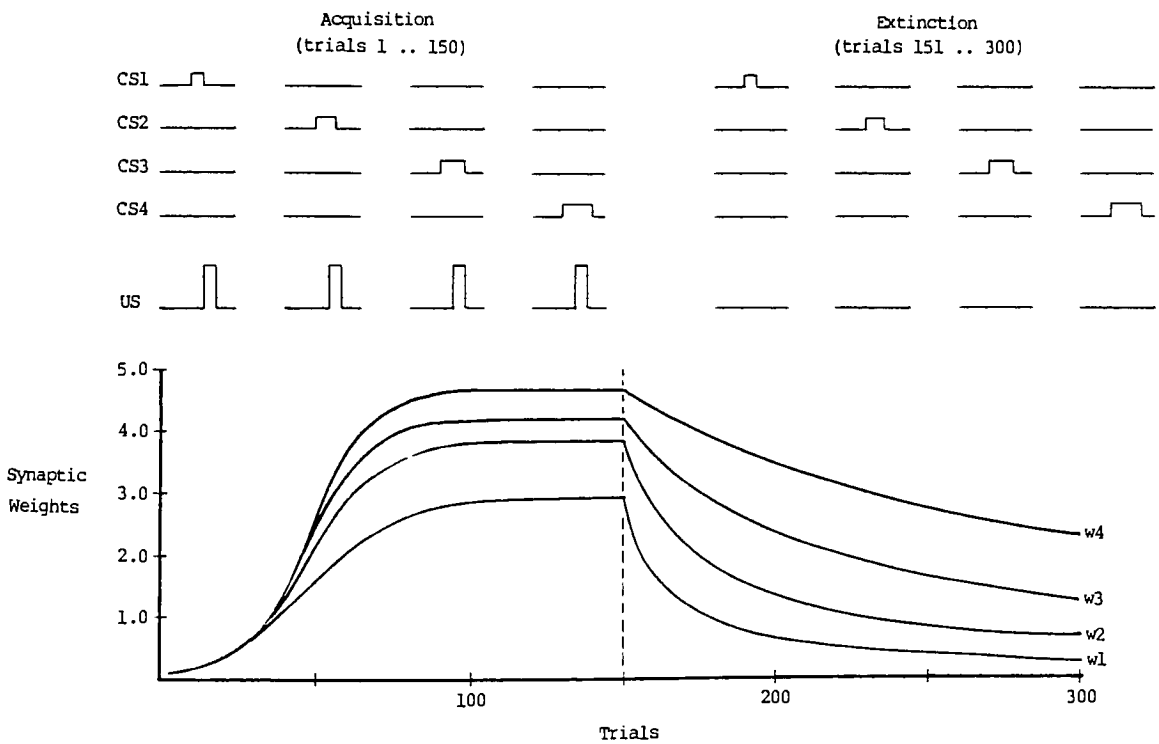


Figure 4.3. The SDR model predicts higher synaptic weight asymptotes as the length of the CS increases (trials 1 through 150). The model also accounts for extinction of the CR in the absence of the US (trials 151 through 300).

Not only is the CS onset-US onset interval important, but the length of the CS also determines relative synaptic weight change differences. Figure 4.3 shows four more runs of the program, using the same constants as before, except for the CSs: this time all CSs have onset at 4; CS1 offset = 6, CS2 offset = 7, CS3 offset = 8, and CS4 offset = 9. Figure 4.3 also shows that the SDR model predicts extinction phenomena (trials 151 - 300).

Figure 4.4 graphs the relationship between the synaptic weight asymptote and

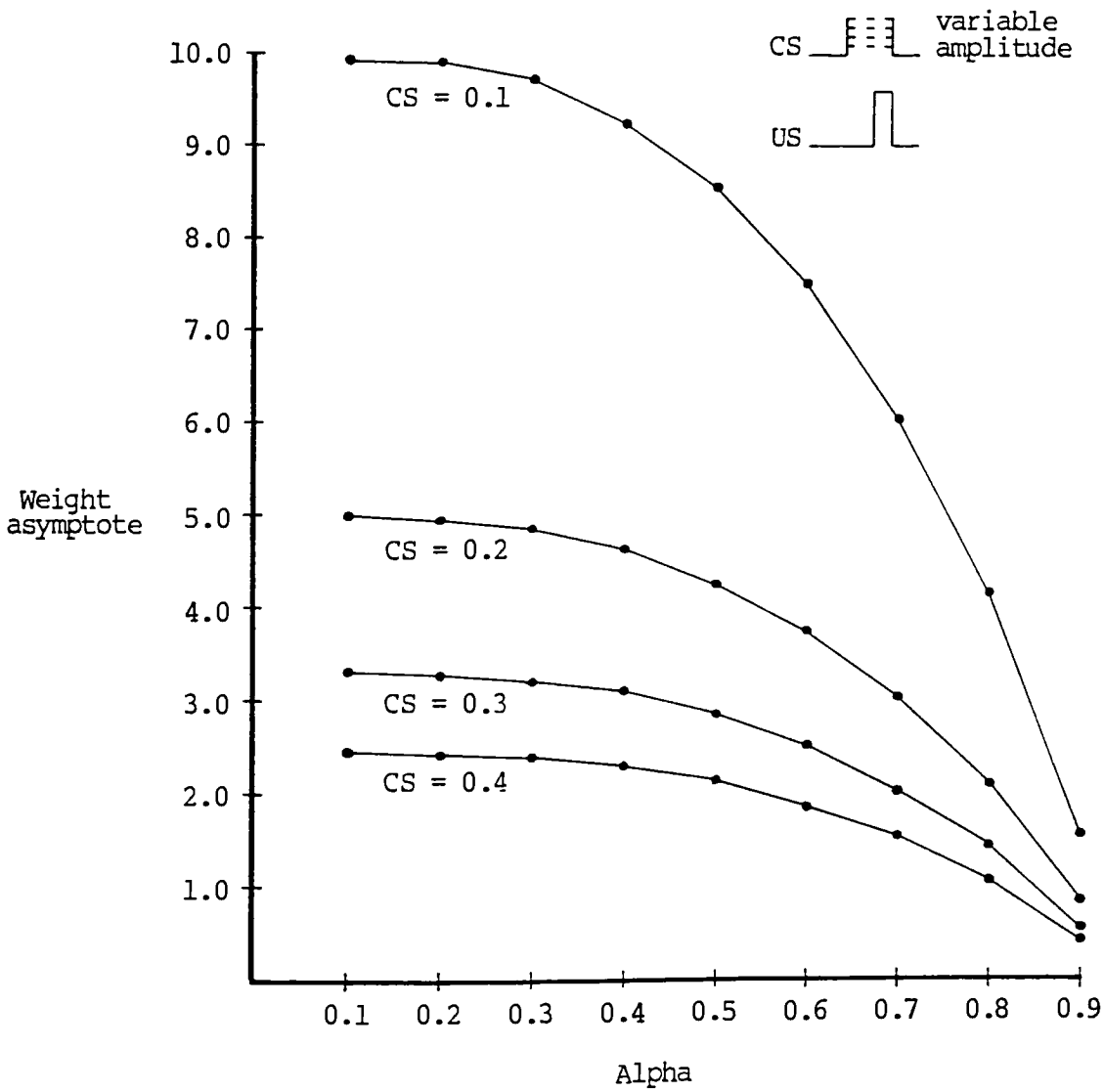


Figure 4.4. Relationship between weight asymptote and the decay rate α for various CS amplitudes, given a fixed US.

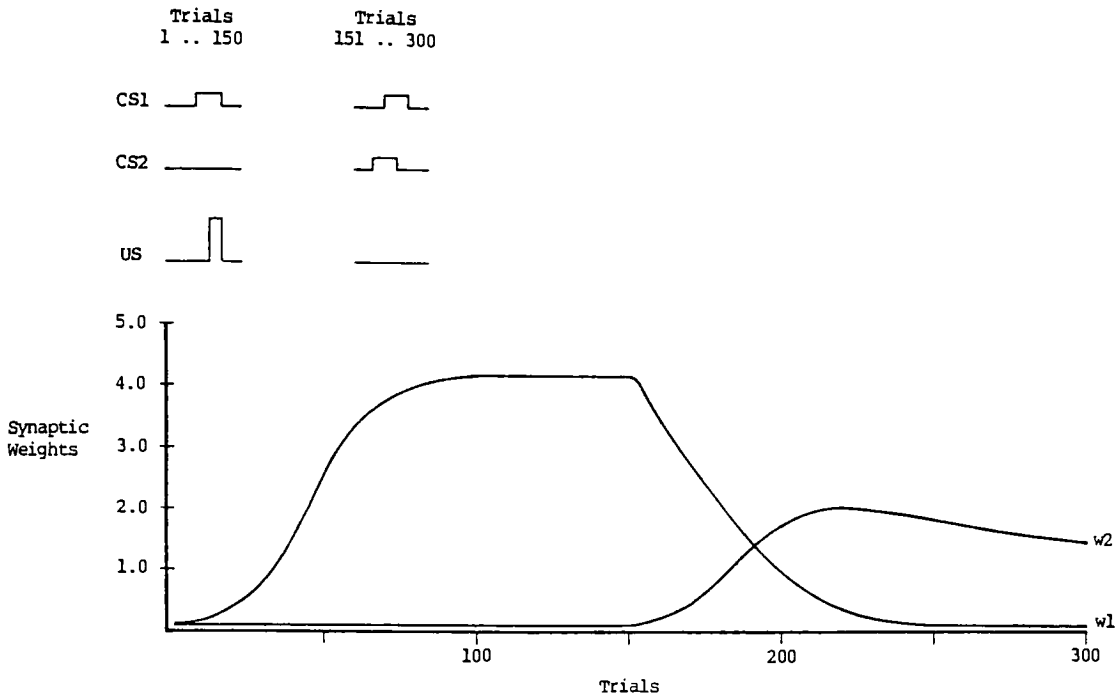


Figure 4.5. The SDR model's predictions of second order conditioning. Note that during trials 151 through 300, the US is absent, and so the response to CS1 is undergoing extinction.

the decay rate constant, α.

4.3.2 Second order conditioning

Figure 4.5 graphs the results of a run of program SDR2.C (see Appendix B), which simulates the phenomenon of second order conditioning. During trials 1 - 150, CS1 is paired with the US, and the synaptic weight w1 rises. During trials 151 - 300, the US is removed and at the same time CS2 is introduced, with onset slightly preceding CS1 onset. CS1 therefore acts as US for CS2. Second order conditioning is rather weak and sometimes ephemeral, because CS1 is undergoing extinction; when w1 finally falls to its lower limit (0.1), CS2 begins to extinguish as well.

4.3.3 Blocking

Figure 4.6 graphs a simulation of the phenomenon of blocking. During the first 150 trials, CS1 is paired with the US, and the synaptic weight (w1) increases. During subsequent trials, a second stimulus, CS2, is compounded with CS1, and both are reinforced by the US. As expected, there is little, if any, gain in synaptic efficacy in CS2, and only a small loss of efficacy in CS1.

4.3.4 Compound conditioning and overshadowing

Figure 4.7 shows a simulation of compound conditioning using program SDR2.C. (See Appendix B.) Both CS1 and CS2 are presented together. Notice that both w1 and w2 remain equal throughout when the amplitudes of CS1 and CS2 are equal, but

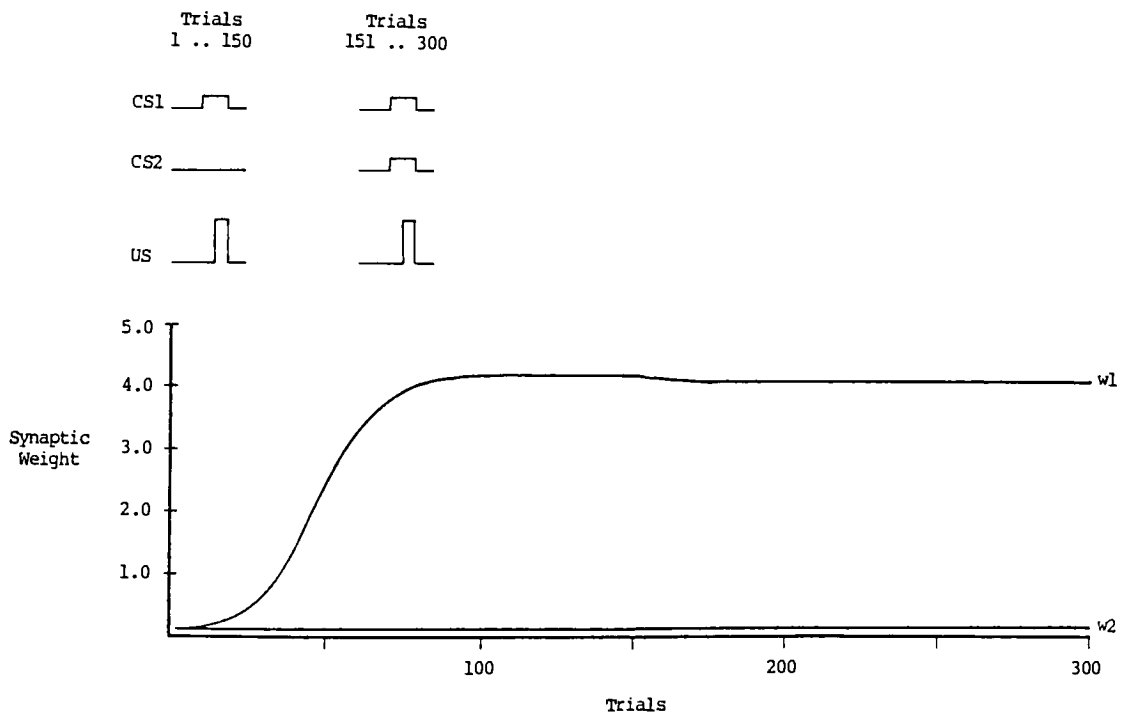


Figure 4.6. The SDR model's prediction of blocking. The second CS, introduced after the first has been conditioned, gains almost no strength.

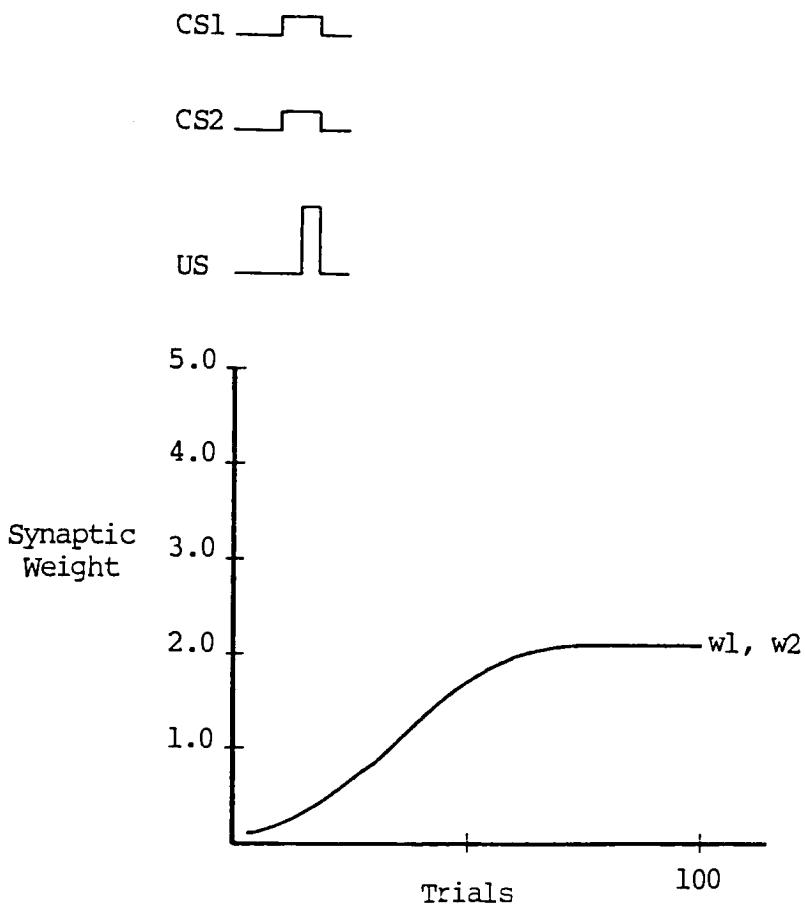


Figure 4.7. Consistently with the Rescorla-Wagner model and empirical results, the SDR model predicts that two equally salient CSs presented simultaneously will generate equal CRs.

the weights do not rise as high as when a single CS is used, everything else being equal (for example, CS4 in figure 4.2). Figure 4.8 is a graph of the same program's output, except that the amplitudes of the two CSs are different. The phenomenon is called overshadowing: a stronger — more salient — stimulus gains proportionally more synaptic efficacy than (i.e., tends to overshadow) a weaker stimulus.

4.3.5 The Wagner-Saavedra experiment

Figure 4.9 shows the results of the Wagner-Saavedra experiment (see section

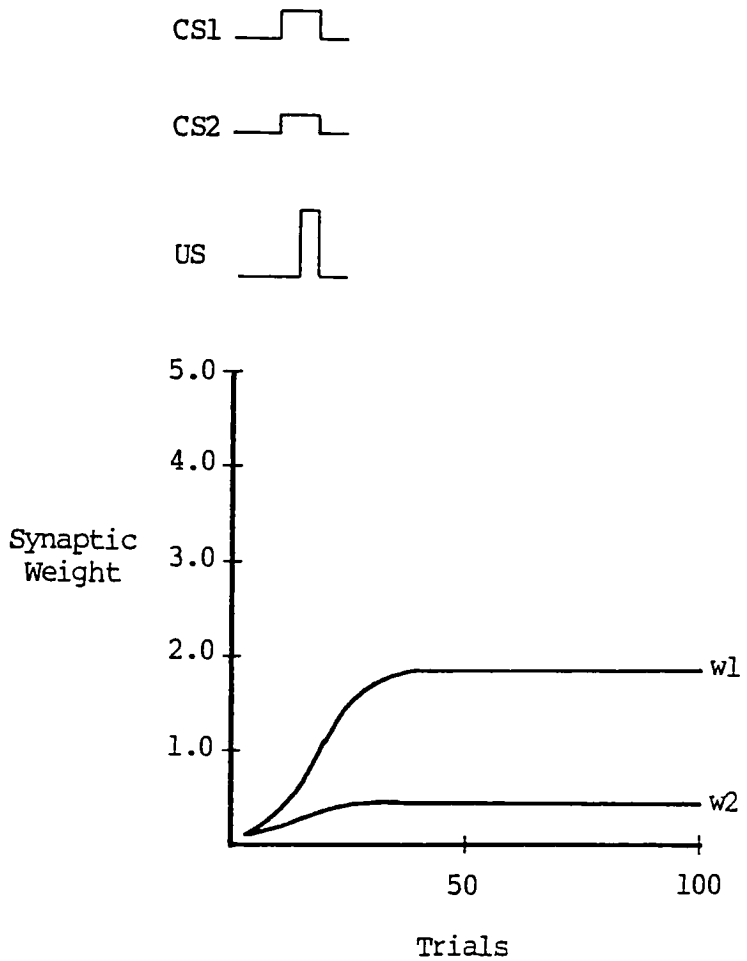


Figure 4.8. The SDR model's prediction in a compound conditioning experiment when one CS is stronger than a second. The first is said to overshadow the second.

3.2) using the SDR model. The results accord well with the graphs shown above in figure 3.5 derived from animal learning experiments.

Figure 4.10 shows the results of a similar experiment. In this case, CS1 is conditioned during the first 150 trials. On the next 150 trials, CS2 is introduced, but in addition, reinforcement trials with both CSs and the US are alternated with CS1 extinction trials. As the graph makes clear, CS2 now tends more and more to be a better predictor of the US, and so CS1's salience decreases.

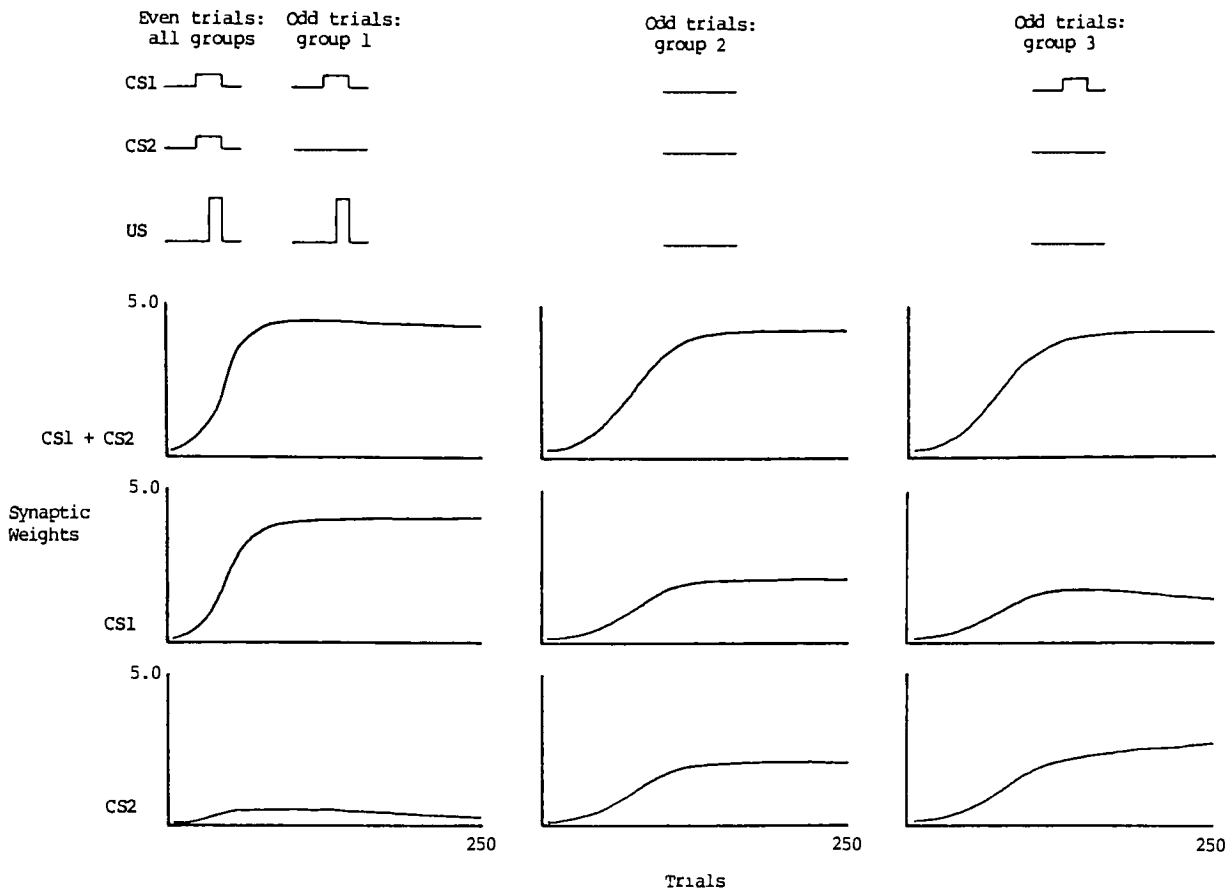


Figure 4.9. The SDR model's prediction for the Wagner–Saavedra experiment. Compare these graphs to those in figure 3.5.

4.3.6 Temporal primacy effects

As mentioned above in section 4.3.1, the closer in time a CS is to the US onset, the more powerful the CS's effects will be — i.e., the higher its asymptote, and the quicker it will approach that asymptote. Nevertheless, an earlier CS will eventually win out over a later CS, as figure 4.11 shows. The SDR model tends to reward the earliest predictor of the US. This temporal primacy effect can even undo the effects of blocking (figure 4.12) and overshadowing (figure 4.13).

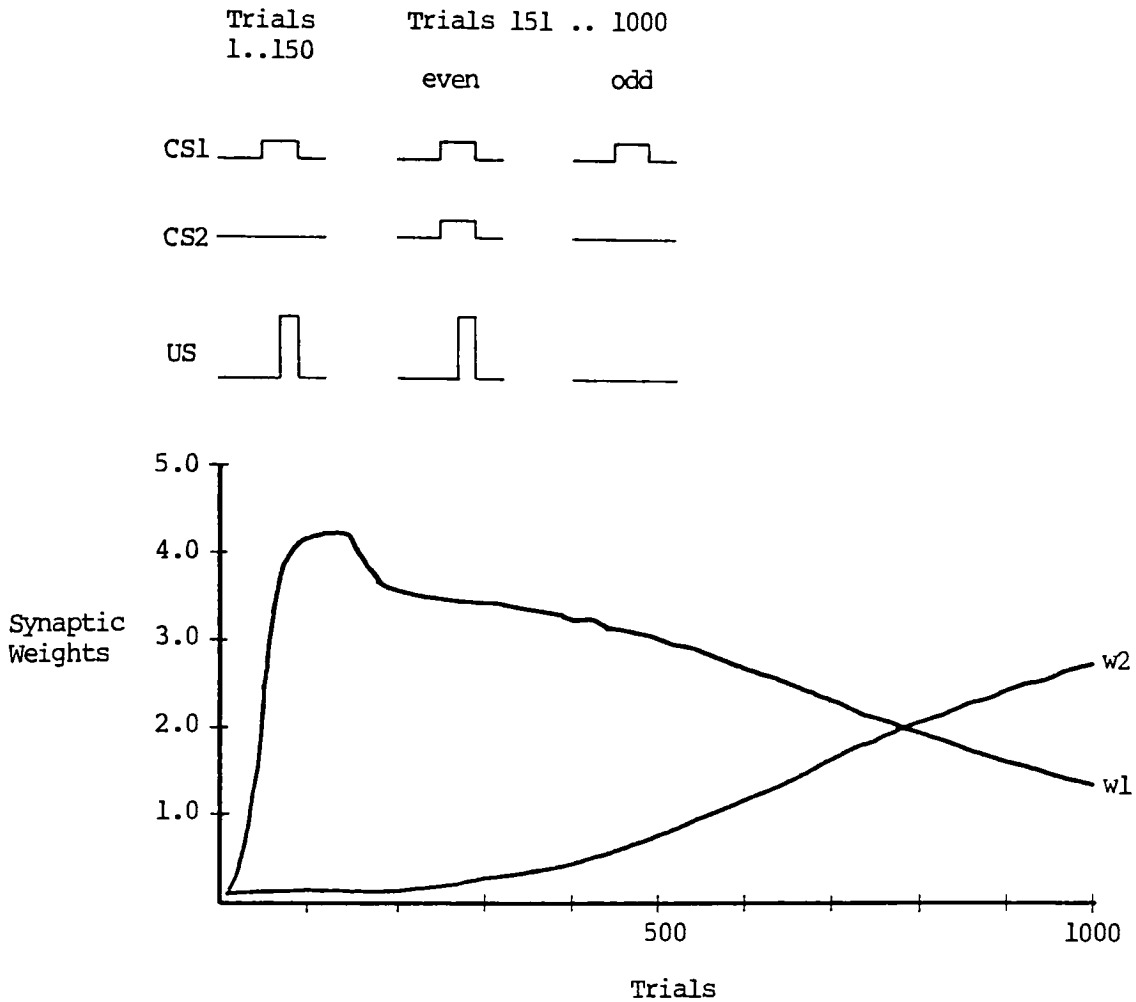


Figure 4.10. After conditioning on CS1, CS2 is introduced on trials alternating with CS1 extinction trials. According to the SDR model, CS2 eventually becomes a better predictor of the US.

4.4 Limitations of the SDR model

In animal learning experiments, as conditioning progresses, the CR begins to appear earlier in the ISI (see section 2.1.6). And if a CR is acquired with a random mix of two different ISIs, the CR will eventually appear with two peaks. But there is no provision in temporal models of conditioning (of which the SDR model is one) to account for this. Instead, the SDR model produces a CR precisely at CS onset; the CR remains constant until CS offset.

Nor does the SDR model take into account the frequency of conditioning

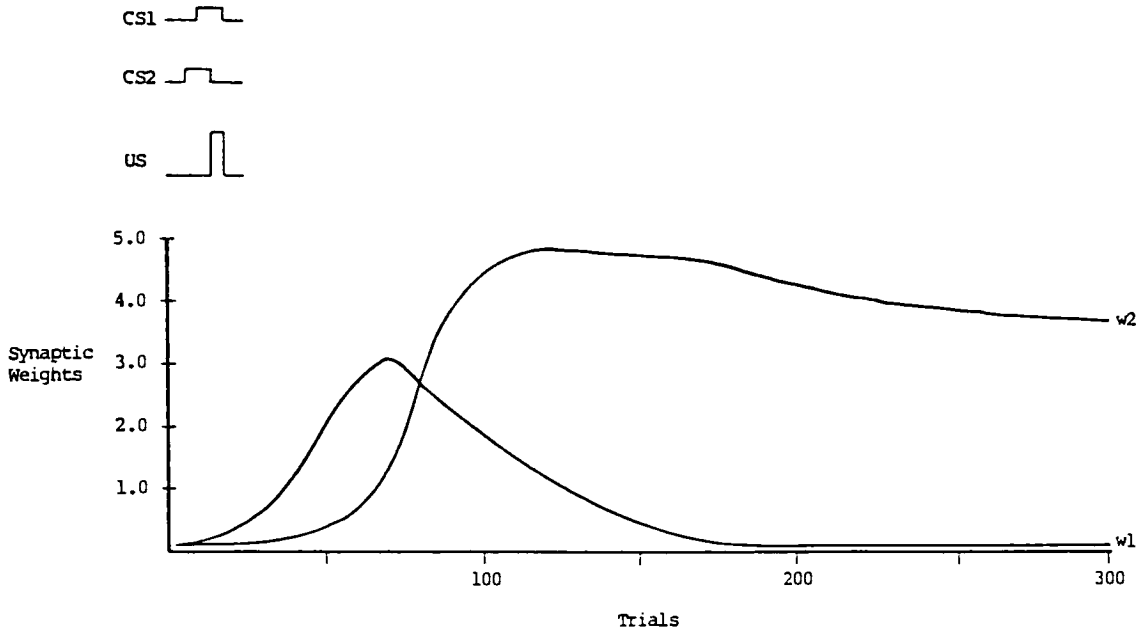


Figure 4.11. The SDR model's predictions about temporal primacy effects. CS1, being temporally closer to the US, will at first gain more weight than an ealier CS. (See figure 4.2.) But CS2, being earlier, is a better predictor of the US and therefore eventually dominates.

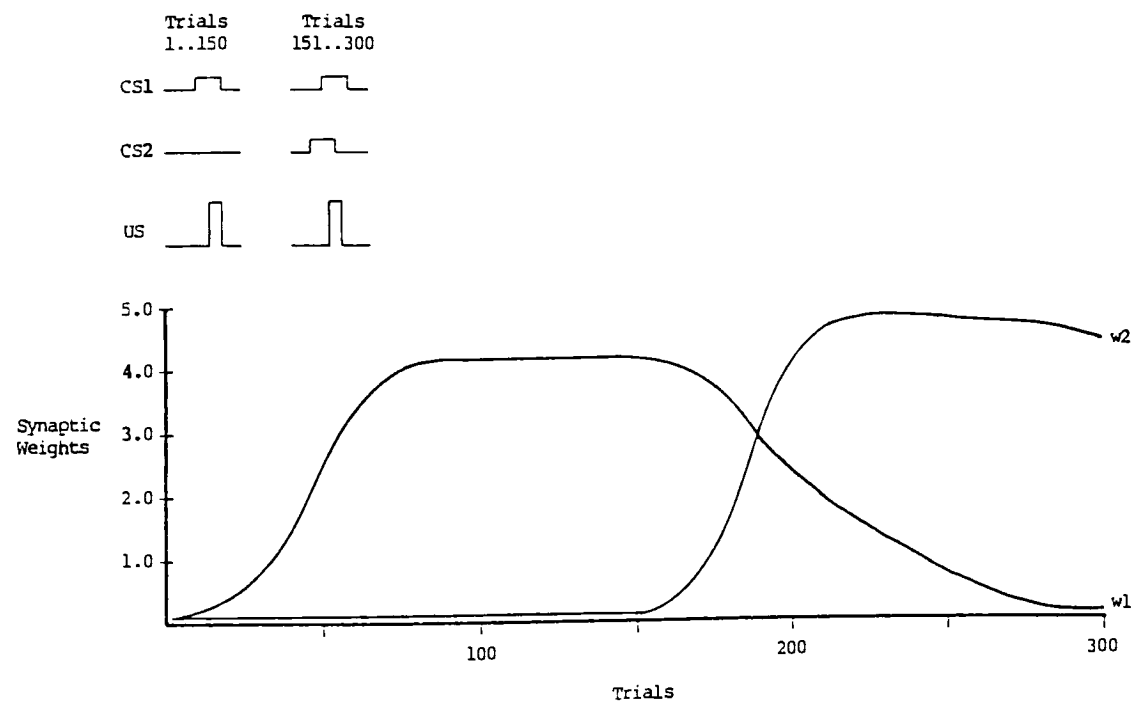


Figure 4.12. The SDR model predicts that temporal primacy can undo the effects of blocking.

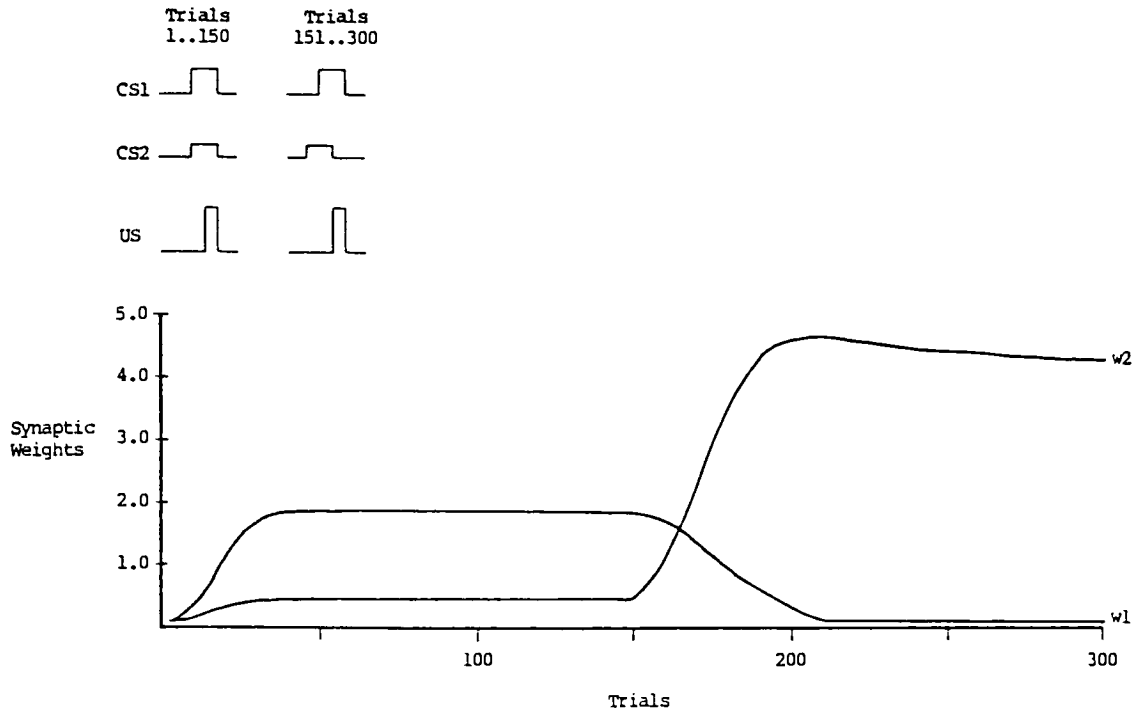


Figure 4.13. The SDR model predicts that temporal primacy can undo the effects of overshadowing.

trials, i.e., the intertrial interval (ITI). Long ITIs seem to promote more rapid conditioning (in terms of number of trials) than shorter ITIs. (Just why this phenomenon appears in animal learning experiments — and whether it is related to some kind of memory consolidation — is an interesting puzzle. The phenomenon also appears in instrumental conditioning, discussed in the next Chapter.)

Sutton and Barto (1990) point out a problem with Klopf-like models (and hence also the present SDR model) employing an eligibility trace which is initiated by CS onset, but which then proceeds independently of the CS: Changes in synaptic weight depend on changes in neuronal output (which may be caused by CS offset or US onset and offset). But extinction of CR occurs in the absence of the US, so in that case it is the CS offset which provides the negative change

in output which causes the loss of synaptic efficacy, depending on the eligibility curve at that moment. But for very long CSs, the eligibility curve will have approached zero (or some small minimum), and so the decrement in synaptic weight will approach zero, which is to say that the model predicts no (or at least extremely slow) extinction for very long CSs. Sutton and Barto say that "the empirical data currently available do not directly contradict this prediction, but they are not supportive of it" (p.514).

Nor does the present SDR model account for reacquisition effects (but see Chapter 6) or spontaneous recovery (of CR following extinction). The SDR model might be made to accommodate spontaneous recovery by positing (at least for some units, if not for all) a tonic response level, such that in the absence of any conditioning, responses will tend to rise to that level over time, even after being forced below that level by extinction trials. Alternatively, inhibitory synapses might play a role; see Chapter 6.

Certain problems of configural learning are beyond the scope of the SDR model. For example, a single SDR unit is incapable of reliably responding to the elements of a compound without also responding to the compound itself. In other words, it cannot solve the "exclusive-OR" problem. (Whether, and in what way, networks of SDR units would perform better is an issue I return to briefly in Chapter 6.)

Finally, there are some kinds of changes in responses which are traditionally discussed in animal learning texts, but which are not always said to participate in learning — or to be a part of the phenomena of associative learning. Muscle fatigue, for example, might be a cause of some behavioral changes during conditioning, but it is not clear that we ought to classify it as a kind of learning, although at the level of the neuron there may be some kind of fatigue

as well. (There is, for example, a limiting rate at which a neuron can fire; the SDR model acknowledges this by assigning the value of 1.0 to an SDR unit's limit.)

Sensitization is said to occur when an input stimulus of small amplitude produces a greater than normal response — or a response where before there was none. The SDR model does not account for this.

Habituation is said to occur when an organism ceases to respond to a stimulus repeated monotonously — the animal "gets used to it". But it is not always said to participate in associative learning, even though it might be classed as a response undergoing extinction; in fact, it shares some of the features of conditioning, including spontaneous recovery (Bower & Hilgard, 1981). Aparicio & Strong (1992) suggest that habituation is integral to any complete model of Pavlovian conditioning, but, unfortunately, is too often neglected.

On the whole, the SDR model performs well in predictions of trial-level and temporal-level phenomena of Pavlovian conditioning. The next issue is to what extent such a model can account for instrumental conditioning effects. That is the subject for the next chapter.

5. Instrumental Conditioning

5.1 Instrumental conditioning phenomena

About the same time that investigators began looking into classical conditioning phenomena, there arose an interest in investigating slightly different, but closely related phenomena, now called instrumental conditioning (or operant conditioning). E. L. Thorndike was an early researcher who tried to provide a purely mechanistic account of learning. What he called "trial and error" learning" or "learning by selecting and connecting" was an alternative to the hypothesis that animals solve problems by thinking through, or reasoning to a solution. On Thorndike's hypothesis, animals select a response from a set of possible responses and then repeat that response if the result of performing it is positive. Initial actions may be accidentally positively reinforcing, but to the extent that reinforcement would follow, to that extent the action would tend to be repeated in similar sensory situations. Useful responses would be gradually "stamped in", and useless responses would be gradually "stamped out". Thus his "Law of Effect":

> Of several responses made to the same situation, those which are accompanied or closely followed by satisfaction to the animal will, other things being equal, be more firmly connected with the situation ...; those which are accompanied or closely followed by discomfort to the animal will, other things being equal, have their connections with that situation weakened." (Thorndike, 1911, p. 244.)

In a close brush with circularity, Thorndike claimed that a "satisfying state of

affairs" is one which the animal tries to maintain (or at least does nothing to avoid), and discomfort, or an "annoying state of affairs", is one which the animal tries to avoid (or at least does nothing to maintain).

In Thorndike (1932) the Law of Effect was revised slightly: rewards always strengthen a connection substantially, whereas punishment weakens it very little or not at all:

> An annoyer which is attached to a modifiable connection may cause the animal to feel fear or chagrin, jump back, run away, wince, cry, perform the same act as before but more vigorously, or whatever else is in his repertory as a response to that annoyer in that situation. But there is no evidence that it takes away strength from the physiological basis of the connection in any way comparable to the way in which a satisfying after-effect adds strength to it." (p. 313.)

Thorndike gave the name the spread of effect to the phenomenon of a reward's influencing not only the connection to which it belonged, but also temporally adjacent connections (before and after), with diminishing effect. And he defined the principle of polarity as the tendency of connections to act more easily in the direction in which they were formed than in the opposite direction.

This appears remarkably similar to classical (Pavlovian) conditioning, which we have already discussed. Yet there is a crucial difference. In the case of classical conditioning, the US follows the CS according to some schedule determined by the experimenter and independently of any behavior of the animal. In instrumental conditioning, on the other hand, the US is contingent upon the animal's behavior. Although such behavior is called a response (or operant, in Skinner's, 1938, terms), it appears before, not after, the stimulus on a given trial. But of course this is not a case of backwards causation or reverse temporal association, or anything quite so exotic. Rather, the response (or operant) is simply behavior within some context, as a result of which the animal

receives feedback; if the feedback is favorable, the operant is strengthened, which is only to say that the animal will have a tendency to behave the same way again in perceptually similar circumstances. And if the feedback is unfavorable, the animal will have less of a tendency to behave the same way again in perceptually similar circumstances.

Skinner (1938) classified responses into two kinds: (1) respondents are said to be elicited by known stimuli, whereas (2) operants are emitted, and need not be correlated with any known stimuli. (Usually an operant acquires a relation to a prior stimulation, in which case it is known as a discriminated operant.) There are two corresponding kinds of conditioning: (1) Type S is conditioning of respondent behavior, that is, it is Pavlovian conditioning, and Skinner suggests that it is limited to autonomic responses. (2) Type R is instrumental — or operant — conditioning; Skinner suggests this may be limited to skeletal behavior ("voluntary systems"). This "two-factor" theory dominated much research for about 30 years (until the late 1960s).

5.2 Are classical conditioning and instrumental conditioning the same?

Bower & Hilgard (1981) say that

> the CS-US interval is an inherent variable in classical conditioning, whereas the response-to-reinforcement interval (the so-called delay of reinforcement) is inherent in the instrumental conditioning paradigm. It is a remarkable fact that the dynamic laws of learning — acquisition, extinction, generalization, and so on — are very similar for the two types of learning situations ...." (p. 65.)

But if various properties of instrumental conditioning so closely parallel those of classical conditioning, we may well wonder if there are any differences that make a difference. Some writers deny any real differences between the two kinds of conditioning. (Hull, 1943; Guthrie, 1952; see also Terrace, 1973.) Here is

Sheffield (1965):

> Every instrumental learning situation is a classical conditioning situation, with reward as US and response-produced cues as US. More specifically, instrumental learning is a differential conditioning situation, in which cues from the general environment are only partially reinforced and cues specific to responses that are in any way incorrect are never reinforced. We can safely deduce that the relation between the correct-response feedback and the response to reward as a US will be exactly that of a classical CR. (p. 317.)

Trapold and Overmier (1972) say that "the fundamental distinction between classical and instrumental conditioning is at best tenuous" (p. 443). (See also Bindra, 1972, and Gormezano, 1965.) Garcia, McGowan & Green (1972) are also skeptical:

> It is difficult enough to distinguish between classical and instrumental conditioning on procedural grounds; it is impossible to maintain the distinction on a functional organismic basis. (p. 15.)

> The principal difference between the two methods lies in the particular segment of the behavioral sequence the investigator selects as his criteria of learning. In classical conditioning he attends to changes in topography and timing of behavioral events which follow reinforcement, and when sufficient modification and stability has occurred he labels these events conditioned responses. In instrumental conditioning, on the other hand, the investigator attends to the timing and topography of behavioral events which precede reinforcement; when sufficient modification and stabilization occurs, he labels these events conditioned responses. Confusion occurs when, as learning proceeds, the classical responses begin to anticipate the reinforcer and thus instrumentally modify its effects upon the animal. (p. 16.)

Can the SDR model be pressed into service for simulating instrumental conditioning? Part of the problem in neural net simulations of instrumental conditioning seems to be the need for preparing a rich enough behavioral repertoire, along with a rich enough behavioral environment, to see instrumental conditioning in action. In contrast, classical conditioning requires a relatively restricted set of conditions. In classical conditioning one observes the relation-

ship among CS, US, CR and UR, and one seeks to isolate them from all else as far as possible. But in instrumental conditioning, some sort of behavior must be seen to emerge from a background of many other possible behaviors. This will therefore require the use of a number of SDR units, linked together in some fashion. Let us examine a relatively simple, single layer network.

5.3 The SDR model in instrumental conditioning experiments: the RAT

RAT is an experimental simulation of operant conditioning. Imagine a laboratory rat in an exercise area marked off in discrete grids or <u>places</u>. Somewhere within the area there is food (a piece of cheese), and the rat is assumed to be hungry enough to find the cheese and eat it. Eating the cheese (or, to simplify matters, merely finding it) will be the occasion for reward (US). The rat is assumed to "know" where it is within the exercise area at all times, and so I give to this simulated rat a set of sensory neurons corresponding to the set of places; each sensory neuron corresponds to one place in the exercise area. Somehow the rat can neither smell nor see nor in any other way sense the cheese in advance of actually stumbling upon it, and so we may have to suppose that the sensory neurons represent some kind of "cognitive map" of the area. (The rat's exercise area is imagined to be a plane, unobstructed grid; but the rat ought to perform just as well in a maze, where occasionally access to some of a place's neighbors is blocked.)

Each time the rat enters a place, its corresponding sensory neuron fires, providing a CS to a set of SDR units, the move neurons (figure 5.1). (This arrangement is vaguely like Scanlon and Johnson's, 1988, PACRAT.) The US affects <u>all</u> move neurons. We may imagine separate pathways from the US source to each SDR move unit, or we may imagine the US to be a change in some chemical concentration surrounding all the synapses. The result in either case is that all move

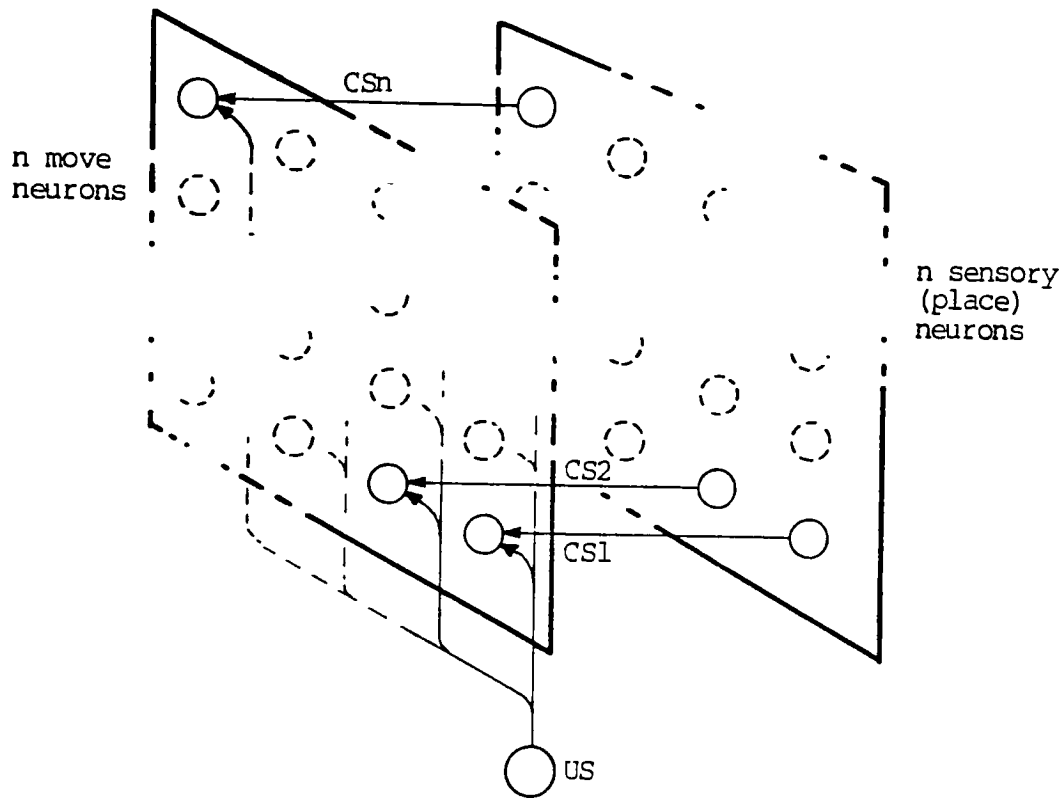


Figure 5.1. RAT simulates a one layer network of SDR neurons (the move neurons), and a layer of sensors which provide the CSs to the move layer. The US — or reward — is presented following the firing of the sensory neuron corresponding to the reward place.

neurons are rewarded when the rat reaches the cheese, but only those neurons corresponding to places where the rat has been (i.e., which have received CSs from sensory neurons) will have their synapses modified, and of those synapses which are modified, the most recently excited neurons will have their synapse weights changed more than those excited earlier, since the eligibility of a synapse decreases with time. After a number of trials (where each trial ends when the rat reaches the cheese), the move neuron corresponding to the location of the cheese will have the highest synapse weight; those neurons corresponding to places exactly one step away from the goal will have slightly lower weights; and so on back to the starting place. If we imagine the rat to be motivated by

the rule, "Move to whichever neighbor place has the highest weight", then it is readily apparent that the rat will be able to make its way to the cheese. (I leave undefined the precise mechanism which senses the highest weight.)

How shall the rat move in case two or more neighbors are equally highly weighted? And how shall the rat move during the first trial, when (presumably) all weights are equally at some minimum (as required by the SDR model)? These issues are easily addressed by having the rat move probabilistically: Let a pseudo-random number be generated within a range partitioned in proportion to each of the (up to four) neighbors' weights. Then the rat will have, not the determination, but rather the tendency to move in the direction of greatest expected reward. The use of such a probabilistic tendency may be supported on

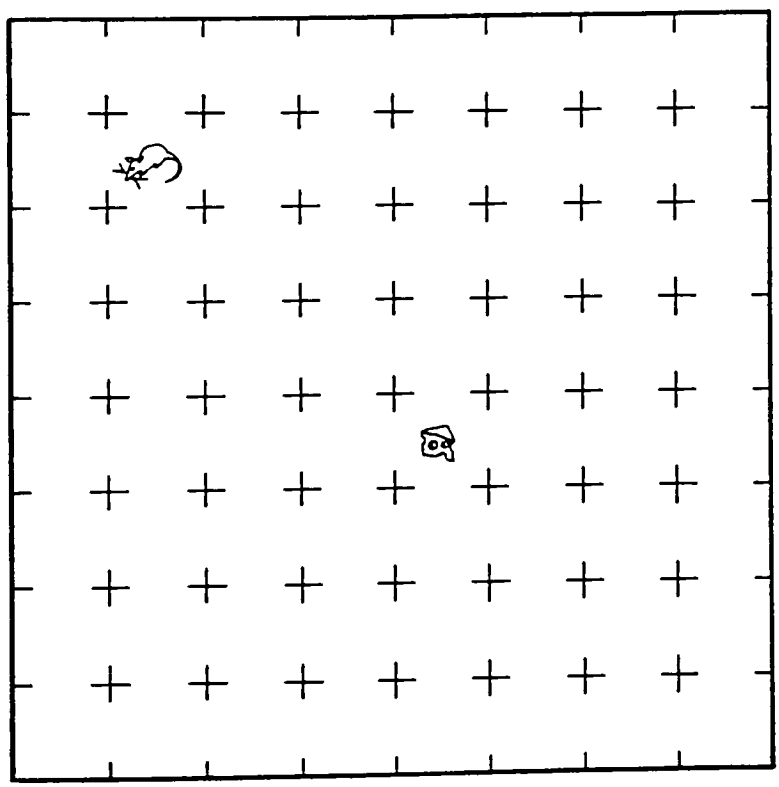


Figure 5.2. A network of SDR neurons is to be conditioned to learn its way to a reward.

other grounds. Presumably, a real rat will have motives other than finding food, and to the extent that such other motives conflict with the search for the cheese, the rat will on occasion move toward other expected rewards. The pseudo-random nature of the move generator will be the pretense at such other, unspecified, motives.

The following describes two experiments using program RAT.C (see Appendix C). Figure 5.2 shows an 8x8 training area with the cheese and the rat. In the first experiment, the rat was started at the same location on each trial. Figure

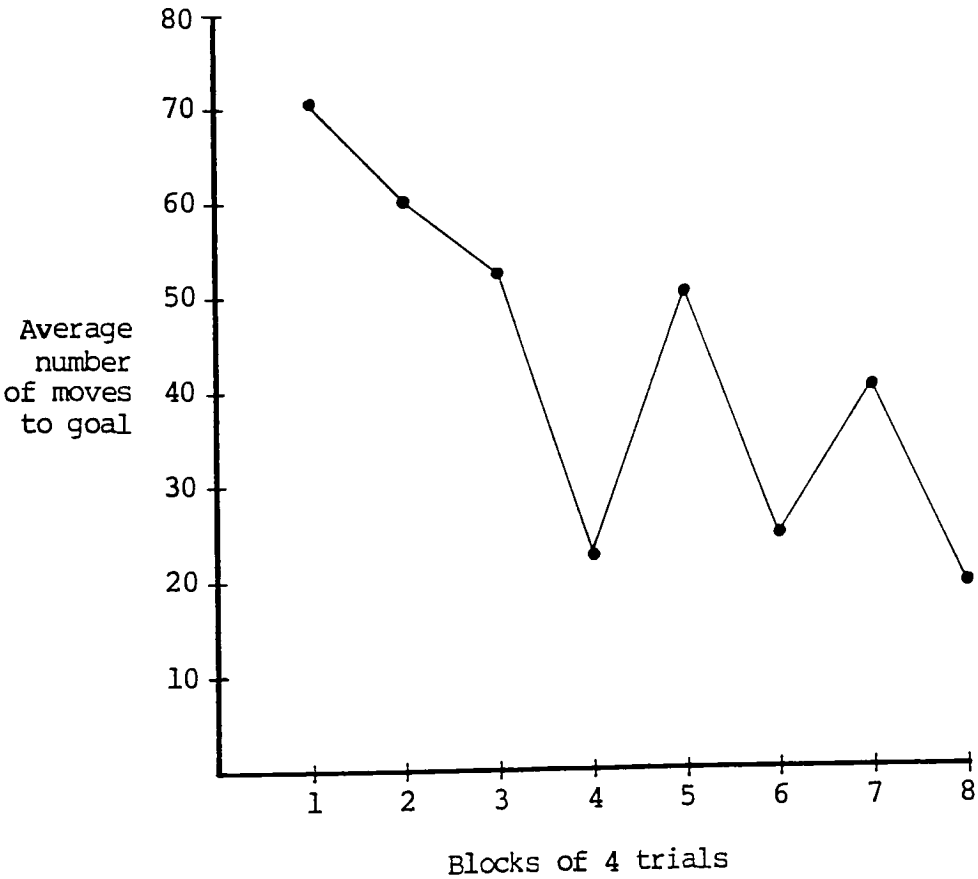


Figure 5.3. During 32 successive rewards, RAT requires fewer and fewer moves to reach the cheese. (RAT was always started from the same place.)

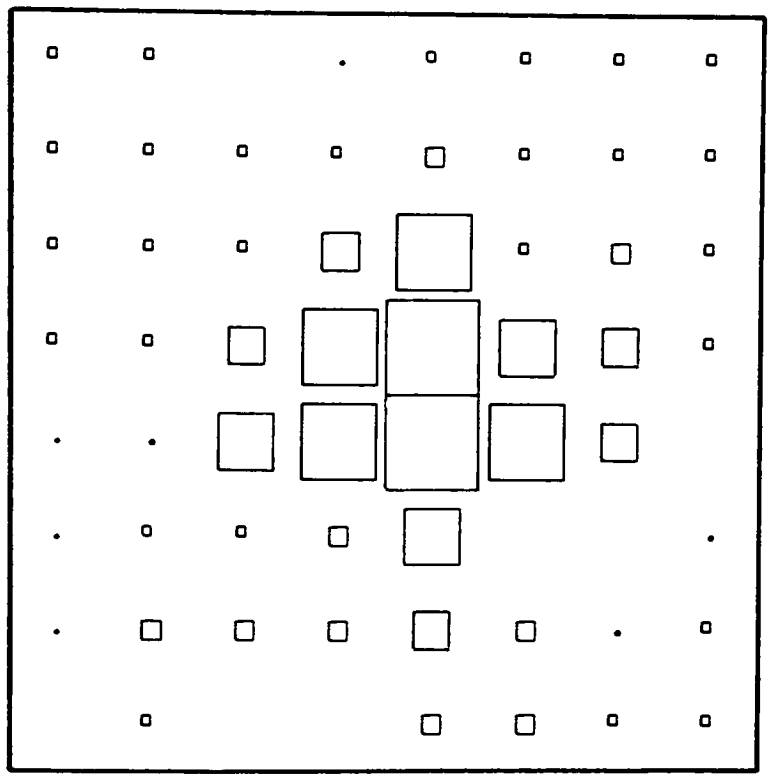


Figure 5.4. RAT has 64 SDR nodes, each fed by a single input. The 64 synaptic weights increase during conditioning, the amount of increase depending on the distance from the goal. Larger squares represent larger weights.

5.3 shows that the number of moves taken to reach the cheese declined significantly over 32 trials, and figure 5.4 shows the relative weights of the 64 modifiable synapses after the final trial.

In the second experiment, program RAT.C was altered slightly to allow six different rats to be trained, each starting at a different Manhattan distance from the goal. (The particular starting cell for each rat on each of its 20 trials was chosen randomly, conditional only on its distance to the goal being the required distance for that rat.) In addition, for each of the six experimental rats, a control rat was started in the experimental rat's 20 starting positions and allowed to move randomly until the goal was reached, but without being

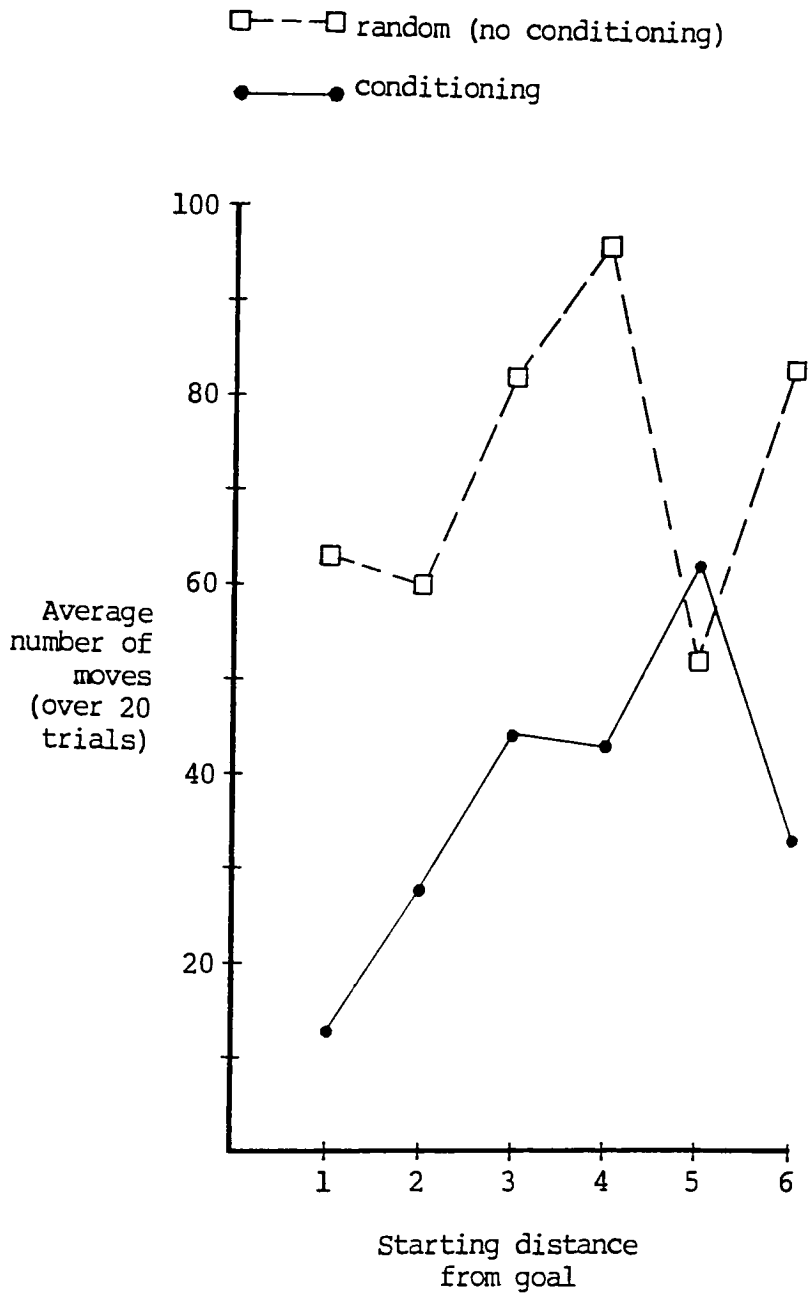


Figure 5.5. Six different rats underwent conditioning, beginning at different distances from the goal. In general, the conditioned rats outperformed unconditioned rats.

rewarded. The control rats, therefore, did not undergo conditioning. Figure 5.5 displays the results.

It is clear from these experiments that a simple, one layer network of SDR

units can function in a simple instrumental learning situation.

5.4 Shaping with Rat

Although the "Law of Effect" -- i.e., instrumental conditioning -- can strengthen only actions which already do occur, more complex behavior can be "shaped" by successive approximations. Suppose, for example, we notice that a cat in a cage will now and then make small leaps into the air, but rarely will a leap exceed five inches in height. Now suppose we reward any leap greater than five inches. Consistently with the "Law of Effect", the cat will eventually be making many leaps greater than five inches, and occasionally a leap may be as high as 10 inches. Suppose now we reward only those leaps higher than 9 inches. Then eventually the cat will be making many leaps over 9 inches.

We may proceed in this manner, conditioning the cat to regularly leap to great heights, limited only by the physical abilities of the cat. Were we initially to reward the cat for very high leaps, we would have to wait quite a long time for even a single such leap, and conditioning would proceed much slower. But by encouraging (rewarding) behavior which already has a tendency to occur -- or, we might say, behavior which is fairly effortless -- we can condition the cat to engage frequently in difficult behavior.

Instead of rewarding the difficult task, we reward an easy task and by small increments condition the animal to more and more complex behavior. Skinner (1938) maintains that most of our complex skills have been synthesized in this fashion.

> Animal trainers are well versed in this method. As a sort of tour de force I have trained a rat to execute an elaborate series of responses suggested by recent work on anthropoid apes. The behavior consists of pulling a string to obtain a marble from a rack, picking the marble up with the forepaws, carrying it to a tube projecting two inches above the floor of the cage, and dropping it inside.

Every step in the process had to be worked out through a series of approximations, since the component responses were not in the original repertoire of the rat. (pp. 339-340.)

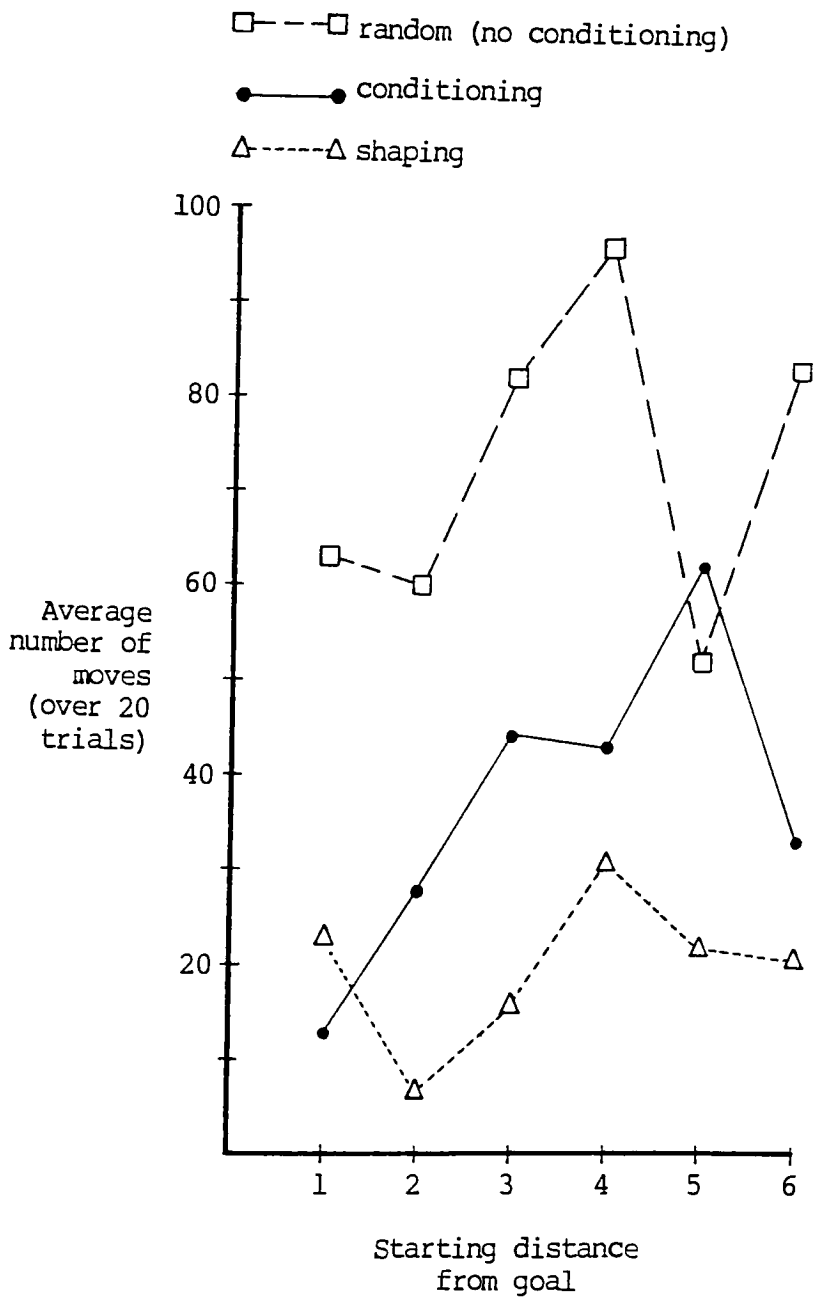


Figure 5.6. Instrumental conditioning using shaping compared to non-shaping and compared to random movement. The non-shaped rats were started at Manhattan distances 1 through 6 from the goal, as were the unconditioned control rats.

In the case of RAT, there is only one task: find the cheese. Nevertheless, we may simulate the process of shaping by initially giving RAT a very easy version of the task: find the cheese when the cheese is in the immediately adjacent place. Once having learned, with some reliability, to solve this easy task, we then begin a new series of trials, this time requiring RAT to find the cheese from two places away. We follow that with the task of finding the cheese from three places away; and so on. Eventually RAT ought to be able to move reliably from the furthest place to the cheese.

Figure 5.6 shows the results (using a version of RAT.C) of shaping a rat compared to conditioning without shaping (six different rats, each started at different distances from the goal) and also compared to six unrewarded control rats.

5.5 A hunger-driven Rat

I proposed the use of a pseudo-random number generator for use with the rat simulation for two reasons. First, it resolves indecision in cases where there is no weightiest neighbor. Second, it substitutes for an indefinite number of unspecified motives (or drives) which a rat might have in addition to its drive to relieve the pangs of hunger. Yet nowhere in the simulation is such a hunger drive explicitly used; rather, the entire simulation is, in effect, a simulation of a hunger drive, assumed always to be active at some constant level. We may now make the hunger drive more explicit.

Let there be an independent modulator of the synaptic weights. The details of how the modulator is generated and what would count as an adequate neuronal circuit are left unspecified. We need only stipulate that the strength of the modulator is directly proportional to the rat's hunger. To simplify matters, we will simply say that hunger <u>is</u> the strength of the modulator. The hunger drive

will begin at some low level and increase over time until some maximum value is reached, or until the rat eats the cheese. Let one effect of such an increase in the drive be a change in the partition for the pseudo-random "weightiest neighbor" decision mechanism by giving a disproportionately larger partition for greater weights. The randomness is still there, but now it is under partial control of the rat's hunger.

The introduction of such a drive leads us to three significant expectations. First, when the rat has a motive — a drive — to learn the location of the cheese, it ought to learn faster than when it does not. We might even postulate that the rat learns nothing at all about the location of the cheese so long as it is not hungry. (Or, to put it another way, reaching the cheese is not a motive unless the rat is hungry.) Second, the hungrier the rat is, the quicker the rat ought to move to the cheese, for any given level of learning. Third, this use of a motivating drive ought to be expandable to include any number of other drives; and to the extent that other drives participate in the simulation, the pseudo-random function (for choosing the weightiest neighbor) can be eliminated, with the result that a rat with a rich set of drives may exhibit a rich set of behavior, even though it behaves deterministically. (For a detailed discussion of drives in animal learning, consult Mowrer, 1973.)

We will say a little more about such a multi-drive beast in Chapter 6. For now, let us take up the first two of the above three expectations. Will a hunger drive provide for faster learning? And will a hunger drive result in fewer steps from a starting place to the cheese, for any given level of learning, and for any given starting place?

Program HUNGER.C (in Appendix D) is a modification of RAT.C. A hunger variable is added which increases (up to some maximum) until the cheese is dis-

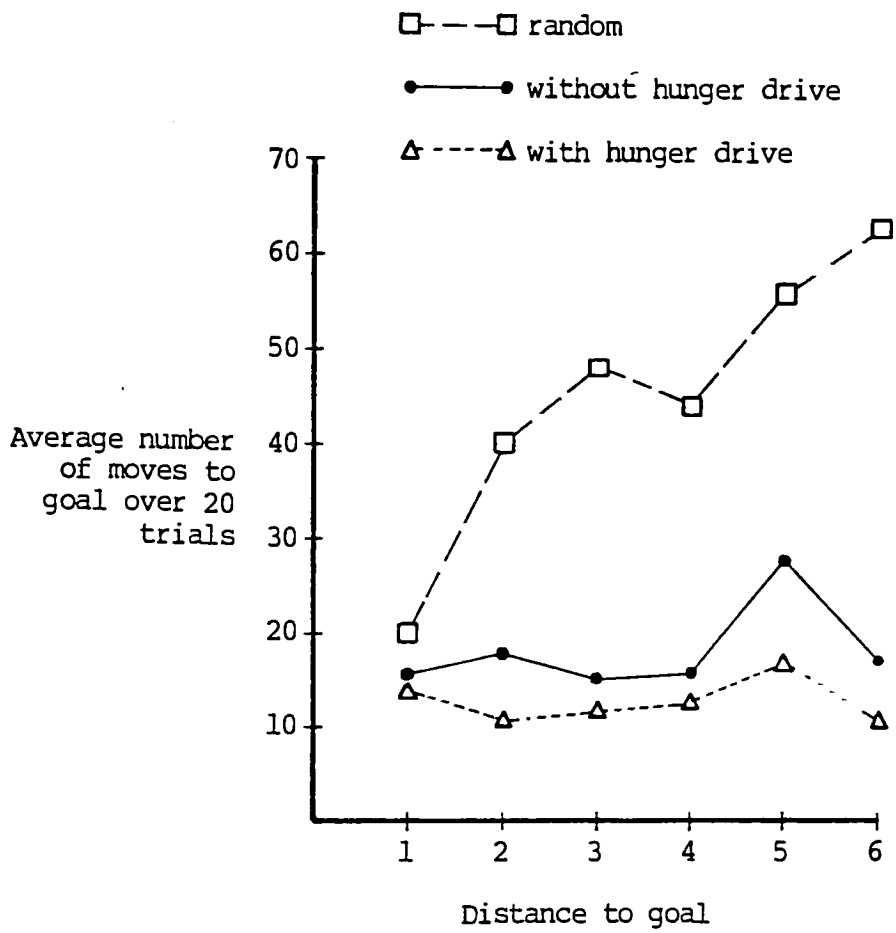


Figure 5.7. Shaping with and without a hunger drive are compared to an unconditioned rat

covered, at which point the variable is reset to zero. The variable influences the weightiest-neighbor decision routine simply by dividing the interval (0, 1) not, as before, in proportion to the four neighbors' weights, but now in proportion to their weights raised to a power represented by the hunger variable (which is kept between 1.0 and 4.0 in program HUNGER.C). The result is that as the rat gets hungrier, it becomes more sensitive to larger neighbors' weights and less sensitive to smaller weights, and will therefore have a growing tendency to follow more closely the "weight gradient" path to the cheese. At the same

time, learning -- which takes place only if the cheese is found -- can now proceed at a variable rate if the hunger variable also influences the amplitude of the US delivered by the cheese. (The presumption made in the program is that hunger increases linearly from 1.0 to 4.0. But a non-linear function can be used to provide an S-shaped curve, as described in section 6.1.)

The results, graphed in figure 5.7, indicate that the hunger drive does indeed encourage learning. But even more interesting possibilities will be mentioned in the following chapter, where still other experiments with the SDR model will be suggested as topics for further exploration.

6. Directions for Future Research

6.1 A multi-drive RAT

If a "hunger" drive can affect the rate of learning, why may there not be other drives as well? A multi-drive rat could have drives for thirst, rest, sex, exploration, and many others. A hunger drive might be more precisely sub-divided into drives for different kinds of nutrition, depending on the needs of the organism.

Each drive might have different rise times and amplitudes (and perhaps different satisfaction rates at a goal state), so that, say, thirst accelerates more slowly than hunger, but has a higher asymptote and slower satisfaction rate — i.e., slower decay time. The heirarchical structure of drives may be made more manifest by making them mutually inhibitory. The result will be that unsatisfied thirst will come to dominate unsatisfied hunger. A rat which is following the weight gradient for food might switch to searching for water; and if fatigue becomes great enough, both hunger and thirst might be suppressed while the rat moves to its nest. As fatigue decreases during the rest, both hunger and thirst will lose inhibition from fatigue and the rat will "wake up" to move off in some motivated direction.

Figure 6.1 depicts such a multi-drive rat. Preliminary investigations (which can best be appreciated in real-time) are encouraging. Figure 6.2 shows the relative strengths of the synapse weights after approximately 8,000 moves in a 6x6 training area with two food goals (to reward hunger), one water goal (to

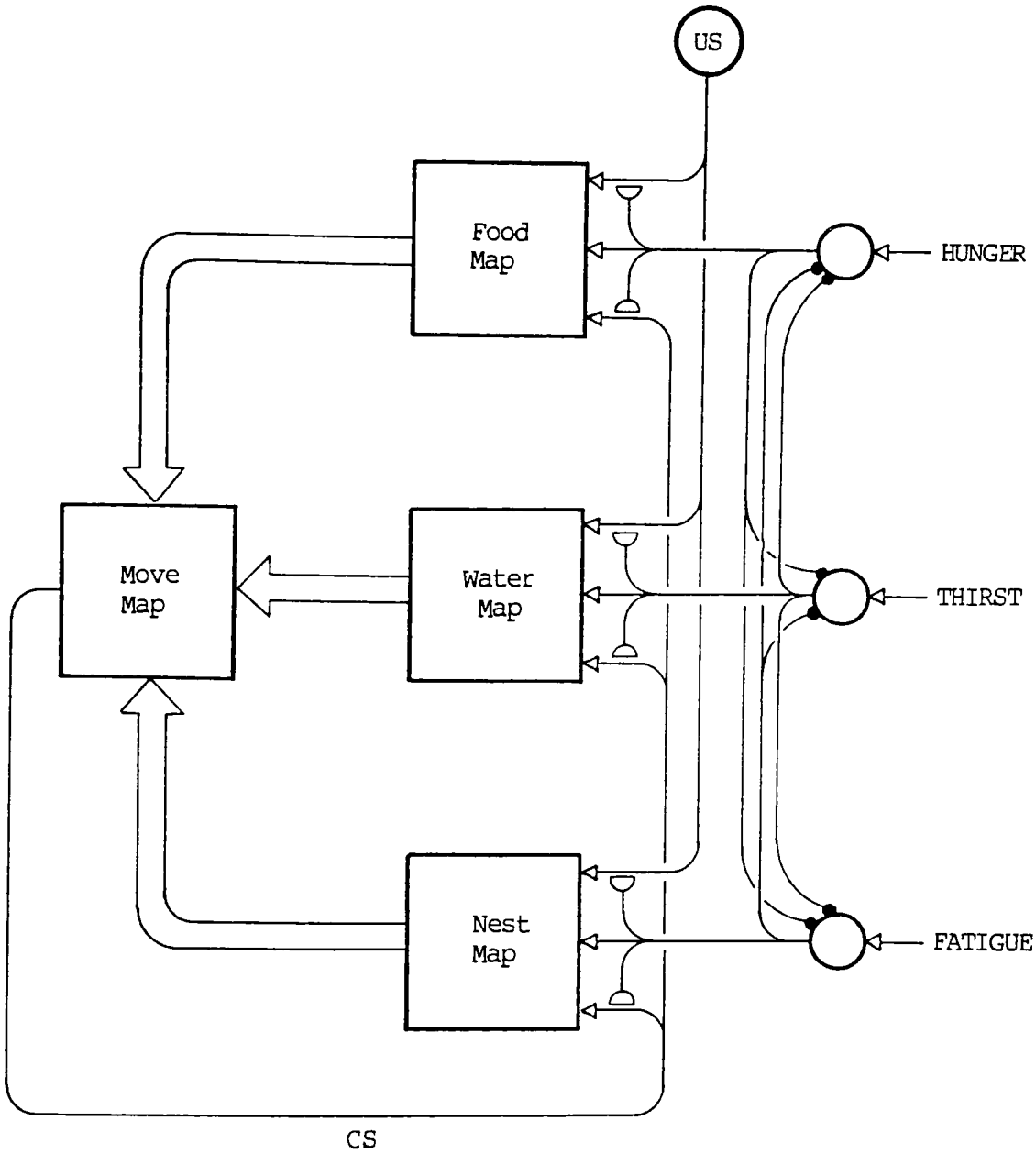
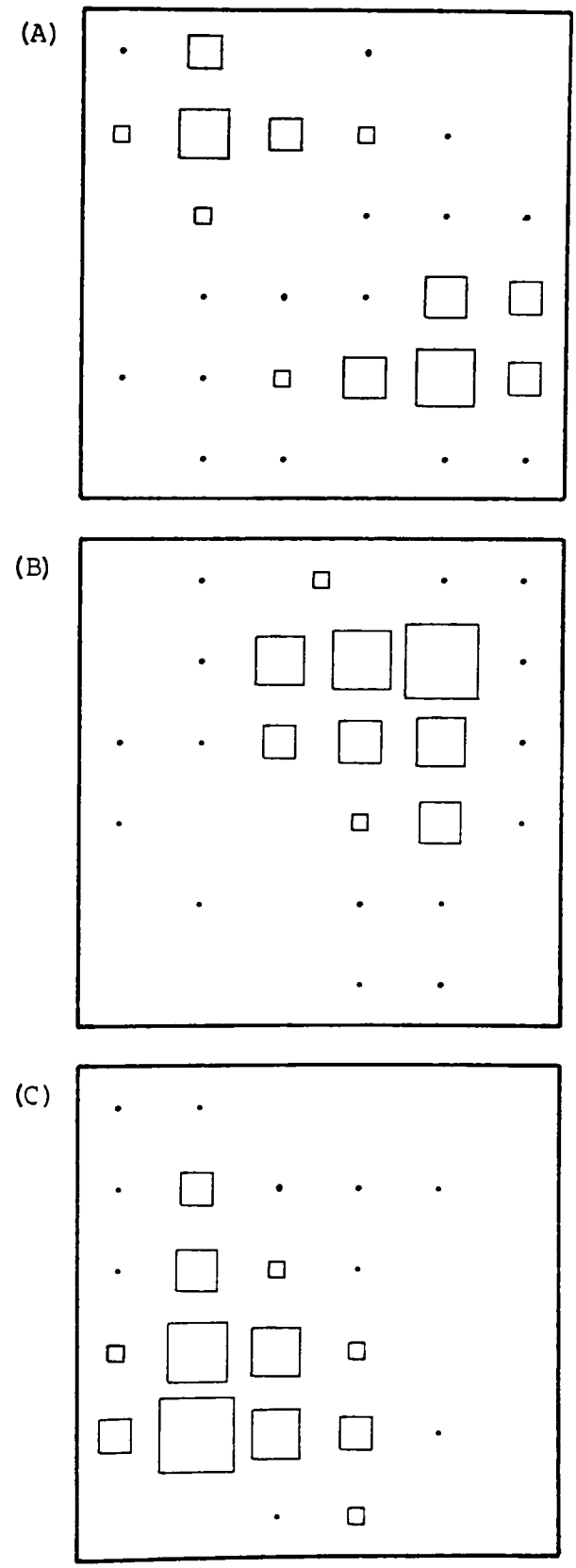


Figure 6.1. Preliminary design of a multiple drive rat. The hunger, thirst and fatigue drives encourage movement toward food, water and a nest, respectively. The drives affect the weightiest neighbor decision routine and the salience of the US (and possibly the various CSs, although it is not clear that affecting the CSs is helpful).

Figure 6.2. The relative strengths of the SDR synapses in a multiple drive rat (in a 6x6 world) in the food layer (a), the water layer (b), and the nest layer (c) after 8000 moves. Larger squares represent larger weights. (Note that there are two food goals.) The fatigue drive grew slower but had a higher asymptote than the other two drives (γ=8.0, η=.06, λ=10). And the thirst drive grew slower but had a higher asymptote (γ=6.0, η=.1, λ=10) than the hunger drive (γ=4.0, η=.2, λ=10).



reward thirst) and one nest goal (to reward fatigue). Drive curves in the multi-drive rat were of the form

$$\text{drive} = \gamma \frac{1}{1 + \exp(-(\eta m - \lambda))}$$

where γ is a constant which determines the amplitude of the drive, η and λ are constants which determine the acceleration of the curve, and m is the number of moves made. USs are to be kept within [0,1], and so if drives are to affect them, drives must be kept within the same range; since γ determines the amplitude of the drive curve, the drive's affect on a US will be

$$\frac{1}{1 + \exp(-(\eta m - \lambda))} \quad .$$

A drive might also be programmed to affect either the learning rate parameter or the decay rate of the SDR eligibility value, so that a well-motivated rat will have a longer useful ISI (interstimulus interval) — i.e., will be able to learn from distant rewards.

6.2 Multi-layered networks of SDR units

SDR units share some features with perceptrons and other simple learning devices, namely, they are limited in the complexity of their ability to distinguish input stimuli (and, hence, limited in their ability to selectively respond). One layer perceptrons, for example, are incapable of solving the exclusive-OR problem. A network with more than one layer presents opportunities for more complex discrimination, provided the credit assignment problem — how to change synaptic weights in an earlier layer so as to produce the required output of a subsequent layer — can be solved (for example, by back propaga-

tion).

SDR units, as I have been using them, are one layer devices, and therefore can be expected to fail the exclusive-OR type problem. Instead of designing multi-layered SDR networks and then tackling the credit assignment problem, the multi-drived RAT simply avoids the issue (for the most part) by means of mutual drive inhibition and selective goal reward: RAT's drives are mutually inhibiting, so that at any time, one drive is (usually) dominant, and reinforcement is given in proportion to the active drive.

There are occasions, however, on which more than one drive is active when a reinforcement is received, and consequently RAT may strengthen a synapse which we know to be inappropriate. On such occasions one drive, having been satisfied, will be reduced, but the second drive will not, and yet its synapse, having been strengthened, will encourage RAT to reenter the (inappropriate) goal state. Nevertheless, such inappropriate reinforcement need not be debilitating, because the first drive, now reduced, also reduces the resulting reward (i.e., the US), and the second drive, now unrewarded at the inappropriate goal, tends to undergo extinction.

Still, multi-layer networks of SDR units ought to be thoroughly investigated. As a beginning, consider the simple arrangement in Figure 6.3, where both inputs to the X unit have modifiable synapses, and the output serves as the input, via another modifiable synapse, to the R unit. Both units receive the US. My first experiments with this lovely couple revealed nothing interesting, and I had already discarded it when I came upon Kehoe's (1992) recent article mentioning just such an arrangement (although for the purposes of discussing a trial-level model of conditioning). Suppose the X unit is conditioned to CS1. Then the X-R connection will also be facilitated, although with some time lag. Now condi-

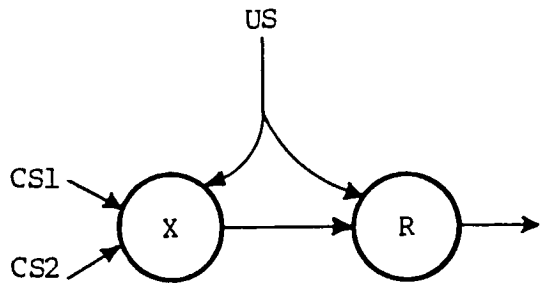


Figure 6.3. A simple pair of SDR nodes. The CS1-X, CS2-X and X-R connection weights are modifiable. This arrangement might exhibit transfer of learning and reacquisition effects.

tion the X unit using CS2. Since the X-R connection has gained efficacy on account of the prior conditioning with CS1, CS2 ought to produce output at R far more effectively at first than CS1 did at first. We might say that the SDR couple has "learned to learn" — that there will be the appearance of a transfer of learning from CS1 to CS2. That is to say, any time an SDR couple learns something, it will thereafter be quicker to learn something else.

Consider what would happen if the response to CS1 is extinguished (by removing the US reinforcement). The CS1-X synapse will loose efficacy faster than the X-R connection. The X-R connection is undergoing extinction as well, of course, but its input (from X) is continually waning, whereas CS1 does not change. Eventually, if the CS1-X synapse weight is allowed to go to zero, then the output from X will be at zero and consequently the X-R connection will cease extinction even while it has some positive weight. (In the SDR model, though, weights are not allowed to fall all the way to zero, but only to some small value. But we may say that the X-R connection weight will practically cease extinction — anyway, the rate of decrease will continue to slow.) If CS1 is now reconditioned (US is represented), the output from R will climb faster than during the first conditioning trials. That is to say, the SDR couple ought to exhibit faster reacquisition than initial acquisition (which is consistent with

empirical results; see section 2.1.2).

6.3 Inhibition

Chapters 4 and 5 dealt entirely with positive (excitatory) synaptic weights. Yet the SDR model allows for negative (inhibitory) weights as well. Equation 4.2 bears repeating here:

$$e_i(t) = \alpha e_i(t-1) + |w_i(t-1)| \min[0, \Delta x_i(t-1)] \qquad (4.2)$$

In addition, the model specifies that $0 < \text{WMIN} \leq |w_i|$, which is to say that weights do not cross zero; excitatory weights remain excitatory, and inhibitory weights remain inhibitory.

What changes in an SDR unit's behavior will occur if we add an inhibitory synapse for each excitatory synapse? (See figure 6.4.) Weight increases occur on the rising edge of the output (following a positive change in input). Weight decreases occur only on the falling edge of the output (following a positive change in input). But the falling edge is temporally farther than the rising edge from the positive input change. Consequently, inhibitory synaptic weights will gain efficacy slower than excitatory weights, as shown in figure 6.5.

Even greater changes in both excitatory and inhibitory synapses are possible if negative USs are allowed. But since the amplitude of a stimulus is a repre-

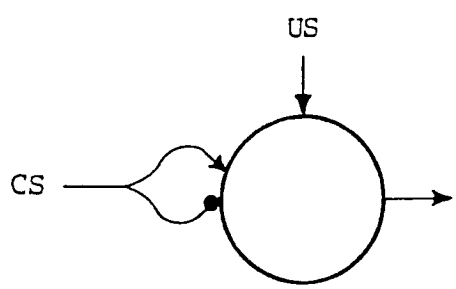


Figure 6.4. The SDR unit with both excitatory (arrow) and inhibitory (circle) synapses.

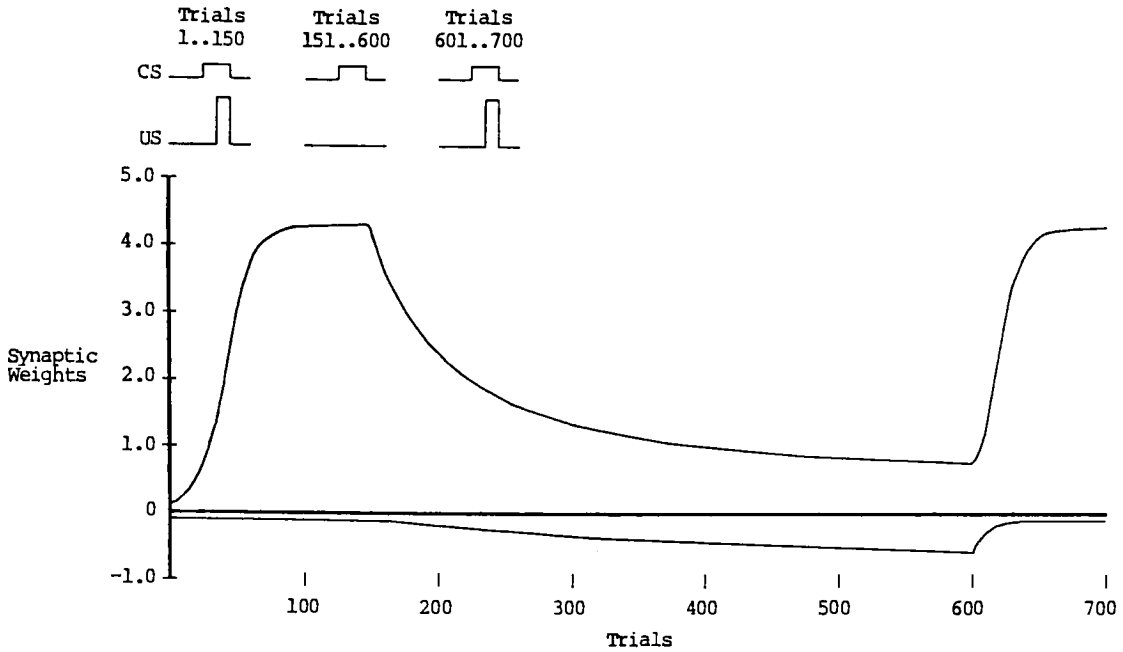


Figure 6.5. Excitatory and inhibitory synapse weights during acquisition, extinction and reacquisition.

sentation of its firing rate, it is not clear how there could be negative rates. So we might suppose instead that there is some level other than zero for inactive USs. Perhaps the inactive US value is 0.5, excitatory (reward) USs range from 0.5 to 1, and inhibitory (aversive) USs range from 0 to 0.5. Or, equivalently, let there be a tonic level of an SDR unit's activity, such that $0 < \text{tonic} < 1$, and the unit's output is tonic when there is no input. This would allow for both positive and negative USs.

All the classical conditioning experiments discussed in Chapter 4 ought to be repeated using the added inhibitory synapses to make sure that no behavior inconsistent with empirical results occurs. Similarly, Chapter 5's experiments ought to be repeated. Perhaps some aversive stimuli (such as poisonous food or a cat) could be added to RAT's world.

6.4 Thresholds

A threshold is sometimes added to artificial neural net models, such that node activation below the threshold results in output of zero. Klopf (1986) explicitly provides for a threshold in his Drive-Reinforcement model, but then effectively discards it by giving its value as zero in his experiments.

So far, nothing at all has been said of threshold values for the SDR model. Perhaps we might say that we have assumed its value to be zero. But suppose it is greater than zero. Might there be a mechanism for changing a threshold value? What effect will that have on the classical conditioning experiments or the instrumental conditioning experiments? Three possibilities come to mind. Suppose an input stimulus is below the threshold and therefore produces no output. If subsequently the threshold is lowered, then what before was too weak to evoke a response will now produce a response. This may be part of what occurs in sensitization. (See section 4.4.) Threshold changes might also participate in changes to the weightiest neighbor decision routine which a hunger (or other) drive produces in RAT. Finally, an SDR couple, described above in section 6.2, might exhibit interesting behavior if threshold modification is introduced (especially if the X output threshold is different from the R output threshold).

In any case, it remains to be discovered under what conditions a threshold change will occur and how best to incorporate those conditions into the equations for the SDR model.

6.5 Conclusion

The SDR model has proven itself competent to handle many of the phenomena of classical and instrumental conditioning. With the additions of inhibitory synapses and threshold modification, perhaps even greater scope can be obtained. But two major issues need to be addressed. First is the phenomenon mentioned in

section 4.4 concerning the length of the ITI (intertrial interval): When the ITI is increased, the number of trials required for conditioning tends to decrease. A related phenomenon occurs in instrumental conditioning, where the strength of conditioning depends on the schedule of reinforcements. One of the ways in which the success of conditioning is measured is by the rate of behavior, e.g., number of bar presses per minute. Skinner discovered quite early that different schedules of reinforcement had a profound effect on both the rate of learning and the resistance to extinction. Less reinforcement does not in general produce less responding. A cat rewarded with a food pellet only every other time it presses a lever, for example, will result in behavior extremely resistant to extinction; reinforcing only one press per minute will produce an increased rate of lever pressing. The schedule according to which reinforcement is given is a complex issue. (Amsel, 1972.) Something, apparently, occurs during the ITI, and the present SDR model seems unable to account for ITI effects.

Second, the exclusive-OR and related problems must be solved. An animal must be able to distinguish, for example, among red round items which are food, red flat items which are poisonous, green flat items which are food, and green round items which are poisonous. (That is, the animal must learn to survive when neither color nor shape is necessary or sufficient to indicate food or poison.) The popular back-propagation method seems to me to be a biologically implausible learning procedure. We have already speculated in section 6.2 that new phenomena might emerge quite readily with even a simple pair of SDR units, so perhaps the nature of the SDR model, as a model of classical conditioning, could offer some plausible alternative when networks of SDR units are fully investigated.

APPENDIX A

```
/*
================================================================
SDR1.C -- Simulation of SDR element in first order conditioning.

A single, excitatory CS is used.


D. B. Suits
May, 24 a.1.

================================================================
*/

#include "stdio.h"
#include "math.h"

#define NAME            "----- Program SDR1.C -----"
#define US_WT           1.0     /* US weight.           */
#define US_AMP          0.7     /* US amplitude.        */
#define MIN_CS_WT       0.1     /* Minimum CS weight.   */
#define CS_AMP          0.2     /* CS amplitude.        */
#define ALPHA           0.4     /* Rate of trace decay. */
#define BETA            1.5     /* Rate of learning.    */
#define MAX_TRIALS      150

#define BEEP            putchar(7)

#define NUM_STR         6  /* Number of cs strings. User chooses one. */

char *cs,
      cs_str[NUM_STR][]  =
                  {"__1111_____",      /* CS0 ('_' = off, '1' = on)  */
                   "___1111____",      /* CS1  */
                   "____1111___",      /* CS2  */
                   "_____1111__",      /* CS3  */
                   "______1111_",      /* CS4  */
                   "_______1111"},     /* CS5  */
     us[] =        "_______11___";     /* US   */

int trial;      /* Counts trials.       */
```

```
float x, prev_x,        /* CS values.   */
      y, prev_y,        /* Node output. */
      elig,             /* Synapse eligibility. */
      weight;           /* Synapse weight.      */

/*
-----------------------------------------------------------------
MAIN --
-----------------------------------------------------------------
*/

main()
{
  int i,c;

    /* Give info on constants, and get user's choice of CS to use.  */

    printf("\n%s\n\n", NAME);
    printf("US_AMP = %3.1f  CS_AMP = %3.1f\n", US_AMP, CS_AMP);
    printf("ALPHA = %3.1f  BETA = %3.1f\n\n", ALPHA, BETA);

    for (i = 0; i < NUM_STR; ++i) {
        printf("cs[%d] = %s\n", i, cs_str[i]);
    }
    printf("us    = %s\n", us);

    while(1) {
        printf("Choose one (0 - %d): ", NUM_STR - 1);
        c = getchar() - '0';
        putchar('\n');
        if ((c >= 0) && (c < NUM_STR))
            break;
        BEEP;
    }

    cs = cs_str[c];

    weight = MIN_CS_WT;

    trial = 0;

    while (trial++ < MAX_TRIALS) {
        prev_y = 0.0;   /* Assume an intertrial interval        */
        prev_x = 0.0;   /*   sufficient to begin afresh         */
        elig = 0.0;     /*   on each trial.                     */

        fire();

        printf("%3d: %f\n", trial, weight);
    }

} /* main() ------------------- */
```

```
/*
-----------------------------------------------------------------
FIRE -- Follow CS and US through one training trial,
        changing weight as appropriate.
-----------------------------------------------------------------
*/

fire()
{
  int t;        /* Time since beginning of trial.       */
  float ux,     /* Weighted input from US.              */
        dy,     /* Change in output.                    */
        dx;     /* Change in input.                     */

    t = 0;
    dx = 0.0;

    while (cs[t]) {
        elig *= ALPHA;  /* Eligibility decreases exponentially. */
        if (dx > 0.0)             /* Only positive changes in   */
            elig += dx * weight;  /*   CS affect eligibility.   */

        x = ((cs[t] == '1') ? CS_AMP : 0.0);
        ux = ((us[t] == '1') ? US_AMP : 0.0);

        y = x * weight + ux * US_WT;
        if (y > 1.0)    /* Output is clipped. Other functions can */
            y = 1.0;    /*   be used, e.g., y = 1 - exp(-y).      */

        dy = y - prev_y;
        dx = x - prev_x;

        weight += BETA * elig * dy;
        if (weight < MIN_CS_WT)
            weight = MIN_CS_WT;

        prev_y = y;
        prev_x = x;

        ++t;

    } /* elihw not done */

} /* fire() ------------------- */

/*
=================================================================
End of SDR1.C
=================================================================
*/
```

APPENDIX B

```
/*
================================================================
SDR2.C -- Higher order conditioning with an SDR unit.

Two excitatory CSs are used.

The program may be adapted to just about any experiment
employing two (or more) excitatory CSs: blocking,
overshadowing, etc.

D. B. Suits
May, 24 a.l.

================================================================
*/

#include "stdio.h"
#include "math.h"

#define NAME            "----- Program SDR2.C -----"
#define US_WT           1.0     /* US weight.           */
#define US_AMP          0.7     /* US amplitude.        */
#define MIN_CS_WT       0.1     /* Minimum CS weight.   */
#define CS_AMP          0.2     /* CS amplitude.        */
#define ALPHA           0.4     /* Rate of trace decay. */
#define BETA            1.5     /* Rate of learning.    */

#define MAX_TRIALS      300
#define SWITCH          151     /* Switch from 1st order to 2nd order */
                                /*   conditioning on this trial.      */

char *cs[2],  /* 2 strings representing CSs. */
     *us,     /* 1 string for the US.        */

     cs_late[] = "_____1111___",  /* CS1        */
     cs_early[]= "___1111_____",  /* CS2        */
     us_str[]  = "_______11___",  /* US         */
     blank[]   = "____________";

int trial;   /* Counts trials.  */

float x[2], prev_x[2],  /* CS values.            */
```

```
      y, prev_y,        /* Node output.           */
      elig[2],          /* Synapse eligibilities. */
      weight[2];        /* Synapse weights.       */

/*
-----------------------------------------------------------------
MAIN --
-----------------------------------------------------------------
*/

main()
{
  int i;

    /* Give info on constants and stimuli. */

    printf("\n%s\n\n", NAME);
    printf("US_AMP = %3.1f  CS_AMP = %3.1f\n", US_AMP, CS_AMP);
    printf("ALPHA = %3.1f  BETA = %3.1f\n\n", ALPHA, BETA);

    printf("cs1 = %s\n", cs_late);
    printf("cs2 = %s\n", cs_early);
    printf("us  = %s\n\n", us_str);

    us = us_str;
    cs[0] = blank; /* Phase A. For phase B, cs[0] = cs_early */
    cs[1] = cs_late;

    weight[0] = 0.0; weight[1] = 0.0;

    trial = 0;

    while (trial++ < MAX_TRIALS) {
        if (trial == SWITCH) {
            cs[0] = cs_early;  /* Go to phase B. */
            us = blank;
        }

        prev_y = 0.0;              /* Assume an intertrial interval */
        for (i = 0; i < 2; ++i) { /*   sufficient to begin afresh  */
            prev_x[i] = 0.0;       /*   on each trial.              */
            elig[i] = 0.0;
        }

        fire();

        printf("%3d: %f %f\n", trial, weight[0], weight[1]);
    }

} /* main() ------------------- */
```

```
/*
-------------------------------------------------------------
FIRE -- Follow CS and US through one training trial,
        changing weights as appropriate.
-------------------------------------------------------------
*/

fire()
{
  int i,
      t;   /* Time since beginning of trial.  */

  float ux,     /* Weighted input from US. */
        dy,     /* Change in output.       */
        dx[2];  /* Change in inputs.       */

    t = 0;
    dx[0] = 0.0; dx[1] = 0.0;

    while (us[t]) {
        ux = ((us[t] == '1') ? US_AMP : 0.0);

        y = ux * US_WT;

        for (i = 0; i < 2; ++i) {
            x[i] = ((cs[i][t] == '1') ? CS_AMP : 0.0);
            y += x[i] * weight[i];
        }

        if (y > 1.0)  /* Output is simply clipped. */
            y = 1.0;

        dy = y - prev_y;

        for (i = 0; i < 2; ++i) {  /* For each synapse ...  */
            elig[i] *= ALPHA; /* Eligibility decreases exponentially.  */
            if (dx[i] > 0.0)                    /* Only positive changes */
                elig[i] += dx[i] * weight[i]; /* affect eligibility.   */

            dx[i] = x[i] - prev_x[i];

            weight[i] += BETA * elig[i] * dy;

            if (weight[i] < MIN_CS_WT)
                weight[i] = MIN_CS_WT;

            prev_x[i] = x[i];
        } /* rof each synapse */

        prev_y = y;
        ++t;
```

```
    } /* elihw stimulus not ended */

} /* fire() -------------------- */

/*
=================================================================
End of SDR2.C
=================================================================
*/
```

APPENDIX C

```
/*
================================================================
RAT.C -- A demonstration of instrumental conditioning using
         SDR units.


D. B. Suits
May, 24 a.1.
================================================================
*/

#include "stdio.h"
#include "math.h"

#define ALPHA           0.95    /* Trace decay rate. */
#define BETA            3.0     /* Learning rate.    */

#define LENGTH          8       /* Width of world, in cells.    */
#define HEIGHT          8       /* Height of world, in cells.   */
#define NUM_CELLS       (LENGTH*HEIGHT)

#define NORTH           0  /* From each cell, rat can move in one */
#define EAST            1  /*   of four directions.               */
#define SOUTH           2
#define WEST            3

#define MIN_WT          0.1     /* Minimum synapse weight.      */
#define CS_AMP          0.2     /* Amplitude of the CS.         */
#define US_AMP          0.7     /* Amplitude of the US.         */

#define GOALX           4       /* World coordinates of         */
#define GOALY           4       /*   the cheese.                */

#define RAT_STARTX      1       /* Rat starts here.             */
#define RAT_STARTY      1

#define MAX_TRIALS      20

char *cs, *us,                  /* Pointers to stimuli strings, below.  */
     cs_on[] =    "_1111_",     /* Time course of various stimuli:      */
     blank[] =    "______",     /*    stimulus is active at '1'.        */
```

```
    us_goal[] =  "___11_";

int ratx, raty,          /* Present rat coordinates.     */
    newx, newy,          /* Rat moves to here.           */
    present_cell,        /* Rat's present cell number.   */
    new_cell,            /* Rat moves to here.           */
    trial,               /* Trial counter.               */
    moves;               /* Move counter.                */

/* Each cell is, in effect, a synapse, and the following arrays
   hold the  ...
*/

float weight[NUM_CELLS], /* ... synapse weight,          */
      elig[NUM_CELLS],   /* synapse eligibility,         */
      prevx[NUM_CELLS],  /* previous CS value,           */
      prevy[NUM_CELLS],  /* previous output value,       */
      dx[NUM_CELLS];     /* and change in input.         */

/*
-----------------------------------------------------------------
*/

main()
{
  int i, j;     /* All-purpose counters.        */

    srand(7);   /* Seed the pseudo-random number generator.     */

    for (i = 0; i < NUM_CELLS; ++i)
        weight[i] = MIN_WT;

    printf("ALPHA = %f BETA = %f\n", ALPHA, BETA);
    printf("%d x %d field.\n", LENGTH, HEIGHT);
    printf("Rat begins at x = %d, y = %d\n", RAT_STARTX, RAT_STARTY);
    printf("Goal cell is at x = %d, y = %d\n", GOALX, GOALY);

    for (trial = 1; trial <= MAX_TRIALS; ++trial) {

        /* Let the intertrial interval be long enough so that
           everything  starts afresh.
        */

        for (i = 0; i < NUM_CELLS; ++i) {
            elig[i] = 0.0;
            prevy[i] = 0.0;
            prevx[i] = 0.0;
            dx[i] = 0.0;
        }

        ratx = RAT_STARTX;      /* Rat starts here,     */
        raty = RAT_STARTY;
```

```
        present_cell = LENGTH * raty + ratx;

        us = blank;

        moves = 0;

        while ((ratx != GOALX) || (raty != GOALY)) {
            fire();
            check_neighbors();          /* Find neighbor with */
                                        /*   highest weight.  */
            ratx = newx;
            raty = newy;
            present_cell = new_cell;
            ++moves;

        }

        printf("Trial %3d: %4d moves\n", trial, moves);

        us = us_goal;

        fire();

        puts("Weights:\n");
        for (i = 0; i < NUM_CELLS; i+=LENGTH) {
            printf("%2d: ", i);
            for (j = i; j < i + LENGTH; ++j)
                printf("%6.4f ", weight[j]);
            putchar('\n');
        }

    } /* rof trials */

} /* main() ---------------- */

/*
------------------------------------------------------------------
FIRE -- Each cell entered is given a CS.
*/

fire()
{
  int c,        /* Counter.     */
      t;        /* Time units along the stimulus.       */

  float x,y,dy; /* CS, node output, and change of output.       */

    for (c = 0; c < NUM_CELLS; ++c) {
        if (c == present_cell)
            cs = cs_on;
        else
            cs = blank;
```

```
        t = 0;
        while (cs[t]) {
            elig[c] *= ALPHA;   /* Eligibility decreases. */
            if (dx[c] > 0.0)    /* Use only positive CS changes. */
                elig[c] += dx[c] * weight[c];

            x = (cs[t] == '1') ? CS_AMP : 0.0;
            y = x * weight[c] + ((us[t] == '1') ? US_AMP : 0.0);
            if (y > 1.0)        /* Output is clipped.   */
                y = 1.0;        /* That, anyway, is the simplest. */
            dy = y - prevy[c];
            dx[c] = x - prevx[c];

            weight[c] += BETA * elig[c] * dy;
            if (weight[c] < MIN_WT)
                weight[c] = MIN_WT;

            prevx[c] = x;
            prevy[c] = y;

            ++t;

        } /* elihw */

    } /* rof each cell */

} /* fire() ---------------- */

/*
-----------------------------------------------------------------
CHECK_NEIGHBORS -- Look at the available neighboring cells (up to
    4) and choose one to move to, based upon a weighted random number.
*/

check_neighbors()
{
  int i, choice;
  float r, sum, extent, tendency[4], partition[4];

    i = present_cell - LENGTH;
    if (i < 0)                                  /* Check North  */
        tendency[NORTH] = 0.0;
    else {
        tendency[NORTH] = weight[i];
    }

    i = present_cell + 1;
    if ((i % LENGTH) == 0)                      /* Check East   */
        tendency[EAST] = 0.0;
    else {
        tendency[EAST] = weight[i];
```

```
    }

    i = present_cell + LENGTH;
    if (i >= NUM_CELLS)                          /* Check South  */
        tendency[SOUTH] = 0.0;
    else {
        tendency[SOUTH] = weight[i];
    }

    i = present_cell - 1;
    if ((present_cell % LENGTH) == 0)            /* Check West   */
        tendency[WEST] = 0.0;
    else {
        tendency[WEST] = weight[i];
    }

    /* Set up a partition of (0..1).    */

    sum = 0.0;
    for (i = 0; i < 4; ++i)
        sum += tendency[i];
    extent = 0.0;
    for (i = 0; i < 4; ++i) {
        if (tendency[i] > 0.0) {
            partition[i] = extent + tendency[i]/sum;
            extent += tendency[i]/sum;
        }
        else
            partition[i] = 0.0;
    }

    choice = -1;        /* Invalid choice.      */
    r = random();
    for (i = 0; i < 4; ++i) {
        if (r <= partition[i]) {
            choice = i;
            break;
        }
    }

    switch(choice) {
        case NORTH:
                new_cell = present_cell - LENGTH;
                newy = raty - 1;
                newx = ratx;
                break;
        case EAST:
                new_cell = present_cell + 1;
                newy = raty;
                newx = ratx + 1;
                break;
        case SOUTH:
```

```
                new_cell = present_cell + LENGTH;
                newy = raty + 1;
                newx = ratx;
                break;
        case WEST:
                new_cell = present_cell - 1;
                newy = raty;
                newx = ratx - 1;
                break;
        default:        /* Never hurts to wear a selt belt, I guess. */
                printf("\nBAD DIRECTION CHOICE: %d \n", choice);
                exit();
    }

} /* check_neighbors() ---------------- */


/*
================================================================
End of RAT.C
================================================================
*/
```

APPENDIX D

```
/*
================================================================
HUNGER.C -- A demonstration of instrumental conditioning using
            SDR units.

A kind of "shaping" is used. The program also incorporates a
"hunger" variable which increases as the number of moves
increases, and which biases the move decision routine even more
in favor of heigher weighted directions.

D. B. Suits

May, 24 a.l.

================================================================
*/

#include "stdio.h"
#include "math.h"

#define ALPHA           0.98    /* Trace decay rate.    */
#define BETA            5.0     /* Learning rate.       */

#define HUNGER_START    1.0     /* Limits and increment (per    */
#define HUNGER_STEP     0.1     /*   each time unit) for the    */
#define HUNGER_MAX      4.0     /*   hunger variable.           */

#define LENGTH          8       /* Width of world, in cells.    */
#define HEIGHT          8       /* Height of world, in cells.   */
#define NUM_CELLS       (LENGTH*HEIGHT)

#define GOALX           (LENGTH/2)      /* Put goal in the      */
#define GOALY           (HEIGHT/2)      /* approximate middle.  */

#define NORTH           0       /* From each cell, rat can      */
#define EAST            1       /*   move in one of four        */
#define SOUTH           2       /*   directions.                */
#define WEST            3

#define MIN_WT          0.1     /* Minimum synapse weight.      */
#define CS_AMP          0.2     /* Amplitude of the CS.         */
```

```
#define US_AMP          0.7     /* Amplitude of the US.          */

#define TRIALS_PER_EPOCH  20  /* Number of times rat must       */
                              /* reach the cheese before        */
                              /* starting at a new place.       */

char *cs, *us,                 /* Pointers to stimuli strings, below. */
     cs_on[] =    "__1111__",  /* Time course of various stimuli:     */
     blank[] =    "________",  /*   stimulus is active at '1'.        */
     us_goal[] =  "____11__";

int ratx, raty,          /* Present rat coordinates.     */
    newx, newy,          /* Rat moves to here.           */
    present_cell,        /* Rat's present cell number.   */
    new_cell,            /* Rat moves to here.           */
    distance,            /* Manhatten distance to goal.  */
    max_distance,        /* Max distance from anywhere.  */
    trial,               /* Trials counter.              */
    moves,               /* Moves counter.               */
    rat_startx, rat_starty,
    total_moves;         /* Total moves counter.         */

/* Each cell is, in effect, a synapse, and the following arrays
   hold the ...
*/

float weight[NUM_CELLS],  /* ... synapse weight,          */
      elig[NUM_CELLS],    /* synapse eligibility,         */
      prevx[NUM_CELLS],   /* previous CS value,           */
      prevy[NUM_CELLS],   /* previous output value,       */
      dx[NUM_CELLS],      /* and change in input.         */

      hunger;             /* The hunger variable.         */

/*
-----------------------------------------------------------------
*/

main()
{
  int i, j, y, x;        /* All-purpose. */

    srand(7);   /* Seed the pseudo-random number generator.     */

    for (i = 0; i < NUM_CELLS; ++i)
        weight[i] = MIN_WT;

    printf("ALPHA = %f BETA = %f\n", ALPHA, BETA);
    printf("CS_AMP = %f, US_AMP = %f\n", CS_AMP, US_AMP);
    printf("HUNGER_START = %f, HUNGER_STEP = %f, HUNGER_MAX = %f\n",
            HUNGER_START,HUNGER_STEP,HUNGER_MAX);
    printf("cs = %s\nus = %s\n", cs_on, us_goal);
```

```
printf("%d x %d field.\n", LENGTH, HEIGHT);
printf("Goal cell is at x = %d, y = %d\n\n", GOALX, GOALY);

/* Determine max Manhatten distance from anywhere to goal.  */

x = (LENGTH - 1) - GOALX;
if (x < GOALX)
    x = GOALX;
y = (HEIGHT - 1) - GOALY;
if (y < GOALY)
    y = GOALY;
max_distance = x + y;

for (distance = 1; distance < max_distance; ++distance) {

    printf("Distance %d\n", distance);

    total_moves = 0;

    for (trial = 1; trial < TRIALS_PER_EPOCH; ++trial) {
        printf("Trial %d: ", trial);

        /* Let the intertrial interval be long enough so that
           everything starts afresh.
        */

        for (i = 0; i < NUM_CELLS; ++i) {
            elig[i] = 0.0;
            prevy[i] = 0.0;
            prevx[i] = 0.0;
            dx[i] = 0.0;
        }

        /* Get starting x and y such that rat starts 'distance'
           units away from goal cell, measured either
           horizontally or vertically.
        */

        while(1) {
            x = random() * LENGTH;
            y = random() * HEIGHT;
            if ((abs(x - GOALX) + abs(y - GOALY)) == distance)
                break;
        }

        rat_startx = x;
        rat_starty = y;

        hunger = HUNGER_START;

        ratx = rat_startx;
        raty = rat_starty;
```

```
            present_cell = rat_starty * LENGTH + rat_startx;

            us = blank;

            moves = 0;
            while ((ratx != GOALX) || (raty != GOALY)) {
                fire();
                check_neighbors();
                ratx = newx;
                raty = newy;
                present_cell = new_cell;
                ++moves;
                ++total_moves;

                /* Hunger variable can increase in any way.     */
                /* Linearly is easy.                            */

                if ((hunger += HUNGER_STEP) > HUNGER_MAX)
                    hunger = HUNGER_MAX;

            }  /* elihw not at goal */

            us = us_goal;
            fire();

            printf("%d\n", moves);

        } /* rof each trial */

        /* At end of each epoch, print out some stats. */

        printf("Ave moves to goal: %f\n",
               (float)total_moves/(float)TRIALS_PER_EPOCH);

    } /* rof each epoch */

    /* Print out weights at end. */

    printf("\nEnding weights:\n");
    for (i = 0; i < NUM_CELLS; i+=LENGTH) {
        printf("%2d: ", i);
        for (j = i; j < i + LENGTH; ++j)
            printf("%6.4f ", weight[j]);
        putchar('\n');
    }

} /* main() ---------------- */

/*
------------------------------------------------------------------
FIRE --
*/
```

```
fire()
{
  int c,  /* Counter.   */
      t;  /* Time units along the stimulus.     */

  float x,y,dy;  /* CS, node output, and change of output.      */

    for (c = 0; c < NUM_CELLS; ++c) {
        if (c == present_cell)
            cs = cs_on;
        else
            cs = blank;

        t = 0;
        while (cs[t]) {
            elig[c] *= ALPHA;
            if (dx[c] > 0.0) /* Use only positive CS. */
                elig[c] += dx[c] * weight[c];
            x = (cs[t] == '1') ? CS_AMP : 0.0;

            /* If hunger is to affect US (as well as weightiest */
            /* neighbor partition in check_neighbors()), then   */
            /* use "US_AMP * hunger / HUNGER_MAX" instead of    */
            /* just "US_AMP" in the expression below.           */

            y = x * weight[c] + ((us[t] == '1') ? US_AMP : 0.0);
            if (y > 1.0)        /* Output is clipped. */
                y = 1.0;

            dy = y - prevy[c];
            dx[c] = x - prevx[c];
            weight[c] += BETA * elig[c] * dy;
            if (weight[c] < MIN_WT)
                weight[c] = MIN_WT;

            prevx[c] = x;
            prevy[c] = y;

            ++t;

        } /* elihw */

    } /* rof each cell */

} /* fire() ---------------- */

/*
-------------------------------------------------------------------
CHECK_NEIGHBORS -- Look at the available neighboring cells (up to
    4) and choose one to move to, based upon a weighted random number.
    In this program, neighbors' weights are disproportioned by
```

```
    raising them to a power represented by the hunger variable.
*/

check_neighbors()
{
  int i, choice;
  float r, sum, extent, tendency[4], partition[4];

    i = present_cell - LENGTH;
    if (i < 0)                                      /* Check North  */
        tendency[NORTH] = 0.0;
    else
        tendency[NORTH] = pow(weight[i], hunger);

    i = present_cell + 1;
    if ((i % LENGTH) == 0)                          /* Check East   */
        tendency[EAST] = 0.0;
    else
        tendency[EAST] = pow(weight[i], hunger);

    i = present_cell + LENGTH;
    if (i >= NUM_CELLS)                             /* Check South  */
        tendency[SOUTH] = 0.0;
    else
        tendency[SOUTH] = pow(weight[i], hunger);

    i = present_cell - 1;
    if ((present_cell % LENGTH) == 0)               /* Check West   */
        tendency[WEST] = 0.0;
    else
        tendency[WEST] = pow(weight[i], hunger);

    /* Set up a partition of (0..1)     */

    sum = 0.0;
    for (i = 0; i < 4; ++i)
        sum += tendency[i];
    extent = 0.0;
    for (i = 0; i < 4; ++i) {
        if (tendency[i] > 0.0) {
            partition[i] = extent + tendency[i]/sum;
            extent += tendency[i]/sum;
        }
        else
            partition[i] = 0.0;
    }

    choice = -1;          /* Invalid choice.       */
    r = random();
    for (i = 0; i < 4; ++i) {
        if (r <= partition[i]) {
            choice = i;
```

```
            break;
        }
    }

    switch(choice) {
        case NORTH:
                new_cell = present_cell - LENGTH;
                newy = raty - 1;
                newx = ratx;
                break;
        case EAST:
                new_cell = present_cell + 1;
                newy = raty;
                newx = ratx + 1;
                break;
        case SOUTH:
                new_cell = present_cell + LENGTH;
                newy = raty + 1;
                newx = ratx;
                break;
        case WEST:
                new_cell = present_cell - 1;
                newy = raty;
                newx = ratx - 1;
                break;
        default:
                printf("\nBAD DIRECTION CHOICE: %d \n", choice);
                exit();
    }

} /* check_neighbors() ---------------- */

/*
================================================================
End of HUNGER.C
================================================================
*/
```

Bibliography

Alkon, D. L. (1983). Learning in a marine snail. Scientific American, 249: 70-84.

Alkon, D. L., Quek, F., & Vogl, T. P. (1989). Computer modeling of associative learning. In D. Touretzky (Ed.), Advances in neural information processing systems, vol. 1 (San Mateo, CA: Morgan Kaufmann), pp. 419-435.

Amsel, A. (1972). Behavioral habituation, counterconditioning, and a general theory of persistence. In A. H. Black & W. F. Prokasy (Eds.), Classical conditioning II: Current research and theory (NY: Appleton-Century-Crofts), pp. 409-426.

Anderson, J. A., & Rosenfeld, E. (Eds.)(1988). Neurocomputing: Foundations of research (Cambridge, MA: MIT Press).

Aparicio, M., & Strong, P. N. (1992). Propagation controls for true Pavlovian conditioning. In D. S. Levine & S. J. Leven (Eds.), Motivation, emotion, and goal direction in neural networks (Hillsdale, NJ: Lawrence Erlbaum Associates).

Barto, A. G., Sutton, R. S., & Anderson, C. W. (1983). Neuronlike adaptive elements that can solve difficult learning control problems. IEEE transactions on systems, man, and cybernetics SMC-13:834-846. Reprinted in Anderson & Rosenfeld (1988).

Bindra, D. (1972). A unified account of classical conditioning and operant training. In A. H. Black & W. F. Prokasy (Eds.), Classical conditioning II: Current research and theory (New York: Appleton-Century-Crofts), pp. 453-481.

Bitterman, M. E. (1965). The CS-US interval in classical and avoidance conditioning. In W. F. Prokasy (Ed.), Classical conditioning: A symposium (New York: Appleton-Century-Crofts), pp. 1-19.

Black, J. E., & Greenough, W. T. (1986). Developmental approaches to the memory process. In J. L. Martinez & R. P. Kesner (Eds.), Learning and memory: A biological view (NY: Academic Press), pp. 55-81.

Bower, G. H., & Hilgard, E. R. (1981). Theories of learning, 5th Edn (Englewood Cliffs, NJ: Prentice-Hall).

Buonomano, D. V., Baxter, D. A., & Byrne, J. H. (1990). Small networks of empirically derived adaptive elements simulate some higher-order features of classical conditioning. Neural networks, 3:507-523.

Buonomano, D. V., & Byrne, J. H. (1990). Long-term synaptic changes produced by cellular analog of classical conditioning in Aplysia. Science, 249:420-423.

Carew, T. J., Hawkins, R. D., Abrams, T. W., & Kandel, E. R. (1984). A test of the Hebb postulate at identified synapses which mediate classical conditioning in Aplysia. Journ. Neuroscience, 4:1217-1224.

Crutcher, K. A. (1986). Anatomical correlates of neuronal plasticity. In J. L. Martinez & R. P. Kesner (Eds.), Learning and memory: A biological view (NY: Academic Press).

Desmond, J. E. (1990). Temporally adaptive responses in neural models: The stimulus trace. In M. Gabriel & J. W. Moore (Eds.), Learning and computational neuroscience: Foundations of adaptive networks (Cambridge, MA: MIT Press), pp. 421-456.

Donegan, N. H., & Wagner, A. R. (1987). Conditioned diminution and facilitation of the UCR: A sometimes-opponent-process interpretation. In I. Gormezano, W. Prokasy, & R. Thompson (Eds.), Classical Conditioning II: Behavioral, neurophyiological, and neurochemical studies in the rabbit (Hillsdale, NJ: Lawrence Erlbaum Associates).

Dreyfus, H. (1970). What computers can't do (NY: Harper).

Dreyfus, H., & Dreyfus, S. (1988). Making a mind versus modeling the brain: Artificial intelligence back at a branchpoint. Daedalus, 117 (1):15-43.

Garcia, J., McGowan, B. K., & Green, K. F. (1972). Biological constraints on conditioning. In A. H. Black & W. F. Prokasy (Eds.), Classical conditioning II: Current theory and research (NY: Appleton-Century-Crofts), pp. 3-27.

Gluck, M. A., Parker, D. B., & Reifsnider, E. (1988). Some biological implications of a differential-Hebbian learning rule. Psychobiology, 16:298-302.

Gluck, M. A., Parker, D. B., & Reifsnider, E. (1989). Learning with temporal derivatives in pulse-coded neuronal systems. In D. Touretzky (Ed.), Advances in neural information processing systems, Vol. 1 (San Mateo, CA: Morgan Kaufmann).

Gluck, M. A., Reifsnider, E. S., & Thompson, R. F. (1990). Adaptive signal processing and temporal coarse coding: cerebellar models of classical conditioning and VOR adaptation. In M. A. Gluck and D. E. Rumelhart (Eds.), Neuroscience and connectionist theory (Hillsdale, NY: Lawrence Erlbaum Associates).

Gormezano, I. (1965). Yoked comparisons of classical and instrumental conditioning of the eyelid response; and an addendum on 'voluntary responders'. In W. F. Prokasy (Ed.), Classical Conditioning: a symposium (NY: Appleton-Century-

Crofts), pp. 48–70.

Gormezano, I. (1972). Investigations of defense and reward conditioning in the rabbit. In A. H. Black & W. F. Prokasy (Eds.) Classical conditioning II: Current research and theory (New York: Appleton-Century-Crofts), pp. 151–181.

Grossberg, S. (1971a). On the dynamics of operant conditioning. Journ. Theor. Biol., 33:225–255.

Grossberg, S. (1971b). Pavlovian pattern learning by nonlinear neural networks. Proc. Nat. Ac. Sci., 68:828–831.

Grossberg, S. (1974). Classical and instrumental learning by neural networks. In R. Rosen & F. Snell (Eds.), Progress in theoretical biology Vol. 3 (NY: Academic Press).

Grossberg, S. (1982). A psychophysiological theory of reinforcement, drive, motivation, and attention. Journ. theor. Neurobiol., 1:286–369.

Grossberg, S. (1987). Studies of mind and brain: Neural principles of learning, perception, development, cognition, and motor control (Boston: Reidel Press).

Grossberg, S. (1990). A neural network architecture for Pavlovian conditioning: Reinforcement, attention, forgetting, timing. In M. L. Commons, S. Grossberg, & J. E. R. Staddon (Eds.), Neural network models of conditioning and action (Hillsdale, NJ: Lawrence Erlbaum Associates), pp. 69–122.

Grossberg, S., & Levine, D. (1987). Neural dynamics of attentionally modulated Pavlovian conditioning: blocking, interstimulus interval, and secondary reinforcement. Applied optics, 26:5015–5030.

Guthrie, E. R. (1952). The psychology of learning (NY: Harper & Row, Rev. Ed.).

Haugeland, J. (1985). Artificial intelligence: The very idea (Cambridge, MA: MIT Press).

Hebb, D. O. (1949). The organization of behavior (New York: Wiley).

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences 79:2554–2558. Reprinted in Anderson & Rosenfeld (1988).

Hull, C. L. (1943). Principles of behavior (New York: Appleton-Century-Crofts).

James, W. (1890). Psychology (Briefer course) (NY: Holt).

Johnson, M. (1987). The body in the mind (Chicago: University of Chicago Press).

Kamin, L. J. (1965). Temporal and intensity characteristics of the conditioned stimulus. In W. F. Prokasy (Ed.), Classical conditioning: a symposium (NY:

Appleton-Century-Crofts).

Kehoe, E. J. (1990). Classical conditioning: Fundamental issues for adaptive network models. In M. Gabriel and J. W. Moore (Eds.), Learning and Computational Neuroscience: Foundations of adaptive networks (Cambridge, MA: MIT Press), pp. 389-420.

Kehoe, E. J. (1992). Versatility in conditioning: A layered network model. In D. S. Levine & S. J. Leven (Eds), Motivation, emotion, and goal direction in neural networks (Hillsdale NJ: Lawrence Erlbaum Associates), pp. 63-90.

Kehoe, E. J., Schrueurs, B. G., & Graham, P. (1987). Temporal primacy overrides prior training in serial compound conditioning of the rabbit's nictitating membrane response. Animal Learning and Behavior, 15:455-464.

Klopf, A. H. (1979). Goal-seeking systems from goal-seeking components: implications for AI. The cognition and brain theory newsletter, 3 (2).

Klopf, A. H. (1982). The hedonistic neuron (New York: Hemisphere).

Klopf, A. H. (1986). A drive-reinforcement model of single neuron function: an alternative to the Hebbian neuronal model. In J. S. Denker (Ed.) Neural Networks for Computing, AIP Conference Proceedings, 151:265-270 (New York: American Institute of Physics).

Klopf, A. H. (1987). Drive-reinforcement learning: a real-time learning mechanism for unsupervised learning. IEEE Conference, Vol 2.

Klopf, A. H. (1988). A neuronal model of classical conditioning. Psychobiology, 16 (2):85-125.

Klopf, A. H., & Morgan, J. S. (1990). The role of time in natural intelligence: Implications of classical and instrumental conditioning for neuronal and neural-network modeling. In M. Gabriel & J. W. Moore (Eds.), Learning and computational neuroscience: Foundations of adaptive networks (Cambridge, MA: MIT Press), pp. 463-495.

Kohonen, T. (1984). Self-organization and associative memory (Berlin: Springer).

Kosko, B. (1986). Differential Hebbian learning. In J. S. Denker (Ed.), Neural Networks for Computing, AIP Conference Proceedings, 151:265-270 (New York: American Institute of Physics).

Lynch, Gary (1987). Synapses, circuits, and the beginnings of memory (Bradford Books).

Malsburg, C. v.d. (1973). Self-organization of orientation sensitive cells in the striata cortex. Kybernetik 14:85-100. Reprinted in Anderson & Rosenfeld (1988).

Mazur, J. E., & Wagner, A. R. (1982). An episodic model of associative learning. In M. L. Commons, R. J. Herrnstein, & A. R. Wagner (Eds.), Quantitative

analysis of behavior: Vol. III. Acquisition (Cambridge, MA: Ballinger).

Miller, R. R., & Spear, N. E. (Eds.) (1985). Information processing in animals: Conditioned inhibition (Hillsdale, NJ: Lawrence Erlbaum Associates).

Moore, J. W., & Blazis, D. E. J. (1989). Simulation of a classically conditioned response: A cerebellar neural network implementation of the Sutton-Barto-Desmond model. In J. H. Byrne & W. O. Berry (Eds.), Neural models of plasticity: Experimental and theoretical approaches (San Diego: Academic Press), pp. 187-207.

Mowrer, O. H. (1973). Learning theory and behavior (Huntington, NY: Robert E. Krieger Publishing Co.).

Newell, A., & Simon, H. (1976). Computer science as empirical inquiry: Symbols and search. Comm. ACM 19:113-126. Reprinted in J. Haugeland (Ed.), Mind design (Cambridge, MA: MIT Press, 1981).

Ost, J. W. P., & Lauer, D. W. (1965). Some investigations of classical salivary conditioning in the dog. In W. F. Prokasy (Ed.), Classical conditioning: A symposium (NY: Appleton-Century-Crofts), pp. 192-207.

Pavlov, I. P. (1927). Conditioned reflexes (G. V. Anrep, Trans.) (London: Oxford University Press).

Pavlov, I. P. (1955). Selected works (Moscow: Foreign Languages Publishing House).

Rachlin, H. (1976). Behavior and Learning (San Francisco: W. H. Freeman).

Razran, G. (1965). Empirical codifications and specific theoretical implications of compound-stimulus conditioning: perception. In W. F. Prokasy (Ed.), Classical conditioning: a symposium (NY: Appleton-Century-Crofts).

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 (New York: Appleton-Century-Crofts), pp. 64-99.

Ricart, R. (1992). Neuromodulatory mechanisms in neural networks and their influence on interstimulus interval effects in Pavlovian conditioning. In D. S. Levine & S. J. Leven (Eds.), Motivation, emotion, and goal direction in neural networks (Hillsdale, NJ: Lawrence Erlbaum Associates), pp. 115-165.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart & J. A. McClelland (Eds.), Parallel distributed processing Vol I (Cambridge, MA: MIT Press), pp. 318-362.

Scanlon, R., & Johnson, M. (1988). Machine emotion. In IEEE Spring Compcon (Washington, D.C.: IEEE).

Schwartz, B., & Lacey, H. (1982). Behaviorism, Science and Human Nature (New York: W. W. Norton).

Searle, J. (1980). Minds, brains and progams, The behavioral and brain sciences, 3. Reprinted in J. Haugeland (Ed.), Mind design (Cambridge, MA: MIT Press, 1981).

Sheffield, F. D. (1965). Relation between classical conditioning and instrumental learning. In W. F. Prokasy (Ed.), Classical conditioning: a symposium (NY: Appleton-Century-Crofts), pp. 302-322.

Skinner, B. F. (1938). The behavior of organisms: An experimental analysis (NY: Appleton Century).

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. (1987). A temporal difference model of classical conditioning. In Proceedings of the 9th annual conference of the cognitive science society (Seattle, WA.).

Sutton, R. S., & Barto, A. (1990). Time-derivative models of Pavlovian reinforcement. In M. Gabriel & J. W. Moore, (Eds.), Learning and computational neuroscience: Foundations of adaptive networks (Cambridge, MA: MIT Press), pp. 497-537.

Terrace, H. S. (1973). Classical conditioning. In G. S. Reynolds, C. Catania & B. Schard (Eds.), Contemporary experimental psychology (Chicago: Scott, Foresman).

Thorndike, E. L. (1911). Animal intelligence (NY: Hafner Press).

Thorndike, E. L. (1932). Fundamentals of Learning (Columbia University Teachers College: Bureau of Publications).

Trapold, M. A., & Overmier, J. B. (1972). The second process in instrumental learning. In A. H. Black & W. F. Prokasy (Eds.), Classical conditioning II: Current research and theory (NY: Appleton-Century-Crofts), pp. 427-452.

Wagner, A. R. (1981). SOP: A model of automatic memory processing in animal behavior. In N. Spear & G. Miller (Eds.), Information processing in animals: Memory mechanisms (Hillsdale, NJ: Lawrence Erlbaum Associates).

Wigstrom, H., & Gustafsson, B. (1983). Heterosynaptic modulation of homosynaptic long-lasting potentiation in the hippocampal slice. Acta Physiol. Scand., 119:455-458.

Young, F. A. (1965). Classical conditioning of autonomic functions. In W. F. Prokasy (Ed.), Classical conditioning: A symposium (NY: Appleton-Century-Crofts), pp. 358-377.