Chapter 10

Reinforcement Learning

10.1 Introduction

Reinforcement learning (RL) studies the ways animals learn about the events in their environment that lead to rewards and punishments, and use this information to choose their actions appropriately. RL is minimally supervised since animals are not told explicitly what actions to perform in various situations, but rather must work this out for themselves on the basis of evaluative information. RL is one of the cases in which it is possible to derive computational, psychological and neurobiological constraints, and in which there are theories that satisfy many of the constraints at all the levels.

Various neural systems have been studied from the perspective of RL. One is the cerebellum, a structure implicated in eyeblink conditioning, in which animals learn to shut their eyes just in advance of disturbances such as airpuffs or weak shocks that are signalled by cues. Another is the primate midbrain dopaminergic system and its targets. We focus on the latter, together with a small fraction of the extensive behavioral data on conditioning in rats. We start by describing the relevant bodies of theory and data and then show in detail how RL addresses them.

Optimizing Adaptive Control

Figure 10.1 illustrates the sort of task that RL methods are designed to solve. Consider the task for a rat in learning how to get as quickly as possible from the start point to the goal (marked '*') where it is rewarded, avoiding the dead ends (marked '–') where it is punished (or at least rewarded less or not at all). The locations in the maze (marked **s**, **f** and **l**) where the rat has to choose between actions are identified by distinctive collections of stimuli such as lights, tones and smells, and it is allowed to learn on the basis of multiple traversals of the maze.

This task poses two particular problems for the rat. The easier one is exemplified by its having to learn whether to go right or left at position **s**. The experimenter knows that every time the rat goes right, it will be rewarded. However, the rat has no way to know that the reward will always be associated with this choice. Indeed, the experimenter may well deliver rewards in a stochastic manner, or change the contingencies after a few trials, forcing the animal to learn to go left instead. Therefore, the rat has to learn which action is better cautiously, taking account of randomness and the possibility of change. It has to balance *exploitation* of the alternative that it thinks is best with *exploration* of possibly even better actions.

The more complicated task faced by the rat is illustrated by its choice at position **l**. In this case, neither action leads to a reward or a punishment immediately, but it has nevertheless to learn to go left. Therefore, it must associate the ultimate reward or punishment with its initial choice of action. Worse, the rat could correctly go left at **l**, but incorrectly go left at **s**, in which case it has somehow to work out that the punishment should be ascribed to the second and not the first action.

This is an example of an optimal adaptive control task. The ultimate requirement is for optimal control, *ie* the best choice of actions at each state (called a *policy* or a *strategy*). However, the rat must also learn the relationship between actions and rewards, hence the requirement for adaptation. There is an extensive body of theory in engineering as to how systems of any sort can come to choose appropriate actions in control problems. An important method on which we focus is called *dynamic programming.* In one RL version of dynamic programming, the animal learns to evaluate how much reward to expect starting from **s**, **f** and **l**. It then uses the comparative expected worths of **s** and **f** to choose the appropriate action at **l**, using methods similar to those that allow it to choose to go right at **s**.

Animal Conditioning

The ability of animals to learn the relationship between stimuli and actions and rewards and punishments, and to choose policies accordingly, is the focus of behavioral psychology. The field is traditionally separated into *classical* or *Pavlovian* conditioning, for which nothing the animal does makes a difference to the delivery of reinforcers, and *instrumental conditioning,* for which the actions of the animal determines what reinforcement is provided. These are extremely rich experimental paradigms that have produced a wealth of data, and an equal wealth of theoretical conflicts. We present a highly simplified account, and discuss some of the conflicts at the end of relevant sections. In particular, although the maze task in figure 10.1 is clearly an case of instrumental conditioning, we will see that

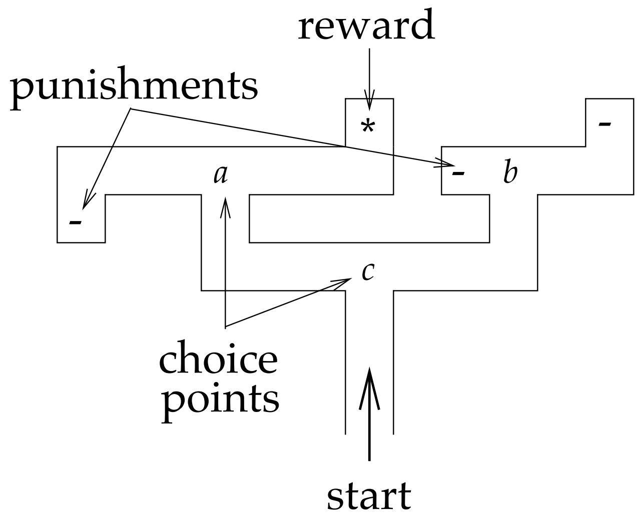

Figure 10.1: A simple maze task to illustrate the temporal credit assignment problem. A model rat has to traverse the maze, getting to the goal as quickly as possible in order to get reward and avoiding the punishments at the positions marked '–'. It can never retrace its steps. States *a*, *b* and *c* are represented by input vectors **s**, **f**, and **l** respectively.

concepts of classical conditioning play a critical role in its solution, and it is on these that we first focus.

In a standard Pavlovian experiment, dogs are repeatedly fed just after a bell is rung. Subsequently, when the bell sounds, the dogs salivate, as if they expect food to arrive. The food is called the unconditioned stimulus. Dogs naturally salivate when they receive food – salivation is thus called the unconditioned response to the unconditioned stimulus. The bell is called the conditioned stimulus, since it is only conditional on the learning that it elicits salivation. The newly minted salivary response to the bell is called the conditioned response. We treat the conditioned responses as a mark of the animal's expectation of the delivery of reward, and so study aspects of the development of such expectations. The references discuss in detail the legitimacy of such abstractions. We generally use lights and sounds as conditioned stimuli.

Animal behavior during classical conditioning is not necessarily adaptive. For instance, one can arrange things such that every time the dog salivates in response to the bell, it is denied food. The development of the salivation response is not impeded by this manipulation. We interpret classical conditioning as an expression of the predictions the animal makes about the delivery of reward. Predictions such as these, for instance, the expectations of the delivery of reward starting from **l**, **s** and **f** in the maze, are a critical component in dynamic programming.

Table 10.1 shows an abstract version of the Pavlovian conditioning task

Name	Set 1	Test
Pavlovian	$l \to r$	$l \stackrel{?}{\to} r$

Table 10.1: Pavlovian conditioning. 'Set 1' is a set of learning trials, which are continued until convergence; 'Test' consists of single or multiple test trials. The $\stackrel{?}{\to}$ in the test set indicates that the association of the conditioned stimulus is tested, revealing, in this case, that the animals expect the reward on presentation of the stimulus.

in a form that will be used to describe other conditioning tasks. 'Set 1' is a whole set of learning or training trials, in each of which the light l is presented in conjunction with the reward r. 'Test' is a test trial, in which the expectation formed about the reward on presentation of the light is probed (indicated by $\stackrel{?}{\to}$). For standard Pavlovian conditioning, the result of learning $l \to r$ during the set of training trials is that the animal comes to expect the reward given the light.

In one version of a standard instrumental conditioning task, rats are rewarded when a particular color light comes on, provided that they press the appropriate lever. Rats quickly learn which lever to press for which color. Learning which lever to push in response to which color is like learning which direction to go in the maze at position **s**, when there are direct rewards and punishments for moving in the correct and incorrect directions. Therefore instrumental conditioning provides a model set of behaviors for understanding how animals perform adaptive control. Similarly, the more complicated tasks of deciding which direction to go at position **l** can be modeled using a combination of classical conditioning (for prediction learning) and instrumental conditioning (for action learning given the predictions).

Dopamine and Predictions of Reward

In a long series of studies by Schultz and his colleagues, monkeys have been trained through instrumental conditioning to perform acts such as moving their arms in response to stimuli such as lights and sounds to obtain food and drink rewards. The activities of cells in the monkey midbrain whose main neurotransmitter is dopamine, are recorded whilst the monkeys learn these tasks.

Dopamine is a neuromodulator, *ie* a neurotransmitter that does not act to depolarise or hyperpolarise cells over short timescales, but rather alters membrane and channel properties, affecting at least excitability and plasticity. There is substantial evidence that dopamine is involved in reward learning. For instance, drugs of addiction such as cocaine and amphetamines act partly by increasing the longevity of dopamine released on target structures such as the nucleus accumbens. Other drugs like morphine and heroin also act through the dopamine system. Another source of evidence is the importance of dopamine in self-stimulation experiments. Rats will readily learn to press levers that cause current to be delivered through electrodes into various areas of their brains. One of the most effective self-stimulation sites is the medial forebrain ascending bundle, which is an axonal pathway. Stimulating this is also likely to cause increased delivery of dopamine to the nucleus accumbens, since the bundle contains many fibers that have connections with the ventral tegmental area (VTA), the region from which Schultz was recording, and which contains the main population of dopaminergic cells that project to the accumbens.

The involvement of dopamine in reward learning is not uncontroversial, since dopamine also plays many other roles, including aspects of attention, and some argue that the results on reward learning are confounded by these attentional effects. Further, reward is a psychologically complicated notion, with specific (*eg* water to slake a thirst; food to quench a hunger pang) as well as generic aspects. Nevertheless, we take the activity of the dopamine cells in the VTA as a critical signal for generic reward learning, and look to it to help understand the neural underpinnings of conditioning behavior.

Figure 10.2 shows the activities of dopamine cells over the course of learning. The figure is based on a reaction time task in which a monkey keeps a finger on a resting key until a light comes on. At this point it has to release this key and press another one to get a fruit juice reward. The reward is delivered $\frac{1}{2}$ second after the second key is pressed. Traces A and B show the activities of a single dopaminergic neuron stimulus-locked to the delivery of the light and reward respectively in early trials (in which the monkey is rewarded for performing almost any appropriate act in response to the light). The histograms show the number of spikes within a short time bin accumulated across trials. It is apparent that the dopamine cells do not have a very high basal firing rate, and that they respond much more significantly to the reward than to the light. Traces C and D show the activities of a different neuron in later trials, by which time the monkey knows the task, but is not yet grossly overtrained (*ie* has not yet performed it so many thousands of times that it is completely automatic). Now the cells fail to respond to the delivery of the reward, but fire significantly to the light. The lower part of these traces show the individual spike rasters and also the time of onset of the arm movement (vertical bars and AMO). It is clear that the firing is locked to the stimulus rather than the response, and there is also a negative correlation between the strength of firing and the delay of onset of the arm movement.

We interpret this activity as reporting a *prediction error* for reward, *ie* an ongoing difference between the amount of reward that is delivered and the amount that is expected. Having prediction error be reported by a system that can control plasticity is appropriate for the classical conditioning task of learning which stimuli are associated with reward; it also turns out to

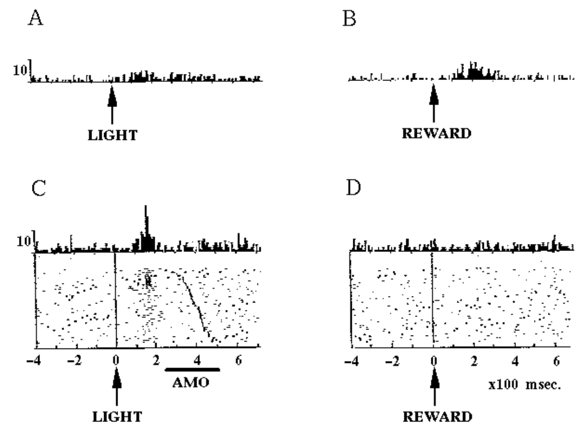

Figure 10.2: Activity of monkey dopaminergic neurons during a reaction time task. Adapted from Schultz (1992).

be appropriate for the instrumental conditioning task of using these predictions to learn the choice of actions. A particular combination of the two is sufficient for solving the maze task.

10.2 Classical Conditioning and Prediction

The Single Conditioned Stimulus

Simple connectionist models have long been used to capture the course of learning in paradigms such as Pavlovian conditioning (table 10.1). Figure 10.3A shows an example of the underlying model, in which the activity of a binary unit u_l is used to represent the presence ($u_l = 1$) or absence ($u_l = 0$) of the light on a trial, weight w_l represents the expectation of the reward given the presence of the light, and, in the simplest case, the actual prediction is given by the activity of unit v as:

$$v = w_l u_l \ . \tag{10.1}$$

One simple way to specify the weight w_l is to minimise the expected square error between the reward r and the prediction v:

$$E(w_l) = \left\langle (r - v)^2 \right\rangle \tag{10.2}$$

where the angle brackets indicate that this is averaged over the presentation of the stimulus and the reward. As we saw in chapter 8, stochastic gradient descent in the form of the delta rule is one way of minimising $E(w_l)$:

$$\begin{aligned} w_l &\rightarrow w_l + \epsilon(r - v)u_l && (10.3) \\ &= w_l + \epsilon(r - w_l) && (10.4) \end{aligned}$$

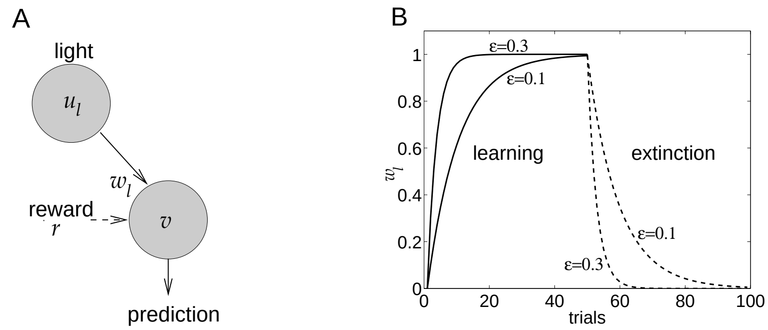

Figure 10.3: Simple model of conditioning. A) Linear prediction model with binary u_l representing the presence of the light permitting prediction v of the reward r through variable weight w_l. B) Exponential acquisition and extinction curves based on the delta rule (equation 10.4). Behavioral acquisition curves are more sigmoidal than this.

where ϵ is a learning rate, which will be interpreted in psychological terms as an *associability* of the light stimulus for the reward. The second form in equation 10.4 follows since u_l is binary. The crucial term in equation 10.3 is the prediction error, $r - v$. As mentioned above, we interpret the activity of the dopaminergic cells in the VTA as reporting a (slightly more complicated) form of this prediction error.

The solid line in figure 10.3B shows the development of the weight w_l over learning trials in which the light and reward are both presented. This shows exponential convergence to the asymptote – experimental learning curves are generally more sigmoidal than this. There are various ways to account for this discrepancy, the simplest of which is to assume a non-linear relationship between the internal expectation v and the observable behavior of the animal.

Minimising $E(w_l)$ provides an account of the ultimate value of w_l; the delta rule provides an account of how this ultimate value is reached. Both these accounts are completely standard in the theory of prediction and, to a good first approximation, capture the results of many conditioning experiments. However, they are inconsistent with the results of many more intricate experiments, which suggest that animals are using more sophisticated prediction models and more sophisticated ways of learning with those models. We lack the space to describe these experiments and the resulting theories, and so just point to some of the more obvious challenges.

The delta rule also accounts for part of the phenomena of partial reinforcement and extinction, shown in table 10.2. If the light is associated with the reward on only some occasions, behavioral measures of the ultimate association of the light with the reward show that it is weaker than if the reward is always presented. This is expected from the delta rule, since the ultimate asymptotic value of w_l is the average reward given that the stimulus is present $E[r|u_l = 1]$. In fact, we see in chapter 11 an interpretation

Name	Set 1	Set 2	Test
Partial reinforcement		$\left\{ \begin{array}{c} l \to r \\ l \to \cdot \end{array} \right\}$	$l \overset{?}{\to} \alpha r$
Extinction	$l \to r$	$l \to \cdot$	$l \overset{?}{\to} \cdot$

Table 10.2: Conditioning paradigms: partial reinforcement and extinction. Partial reinforcement involves just a single set of learning trials (which, for convenience, is labeled 'Set 2'). On some fraction of the trials, the light is presented just before the reward ($l \to r$); on the remainder, the light is shown without the reward ($l \to \cdot$). This results in slower learning and a lower terminal expectation $0 < \alpha < 1$. For the case of extinction, there are two different sets of learning trials. During the first set, the $l \to r$ contingency is presented on each trial; during the second set, the l is presented alone.

of the goal of minimising $E(w_l)$ in terms of estimating the full probability density (or distribution) of $P[r|u_l = 1]$ under a particular statistical model of uncertainty. This statistical interpretation underlies more recent statistical analyses of conditioning. Similarly, if the reward ceases to be given in conjunction with the stimulus, then, under the delta rule, the weight w_l will tend to 0 and, as observed experimentally, the animal will cease expecting the reward following presentation of the stimulus. The dotted line in figure 10.3B shows the course of extinction in the model.

Multiple Conditioned Stimuli

Particularly revealing is the case when there are multiple possible predictive stimuli, for instance both a light and a sound. The extension of the simple model is shown in figure 10.4. If the joint prediction of the model is:

$$v = w_l u_l + w_s u_s = \mathbf{w} \cdot \mathbf{u} \ , \tag{10.5}$$

then using the delta rule to minimise the same form of prediction error $E(\mathbf{w})$ gives as a learning equation the full delta rule (Widrow & Hoff, 1965; Rescorla & Wagner, 1973):

$$\mathbf{w} \to \mathbf{w} + \epsilon (r - v) \mathbf{u} \tag{10.6}$$

Table 10.3 shows three experiments designed to test learning in this case. There are various ways of establishing a negative connection between stimuli and rewards. One of the more reliable is labeled 'inhibitory' – alternating trials in which the light is shown in conjunction with the reward, and trials in which the light and the sound are delivered together in the absence of reward. In this case, the sound becomes a conditioned inhibitor, which, under one interpretation, means that it actively predicts the *absence* of reward. This can be demonstrated by presenting a third stimulus that itself predicts reward, and showing that this prediction is reduced when

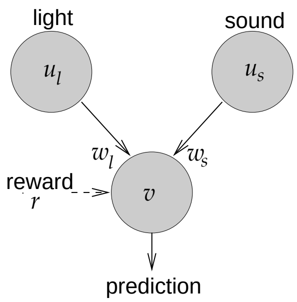

Figure 10.4: Simple prediction model for multiple stimuli.

the new stimulus is presented in conjunction with a sound. This emerges naturally from the task of minimising $E(\mathbf{w})$; weight w_l has to be positive to predict the reward in the upper set of trials, and weight w_s has therefore to be negative so that the net prediction in the lower set of trials is 0.

In all the paradigms so far, there has only been one set of values for the weights that minimises $E(\mathbf{w})$. There is also a substantial body of experiments testing cases for which many combinations of values of the weights lead to perfect predictions, and therefore probing the way that predictions are shared between the weights and the effects of path dependencies. Overshadowing tests the former – if two stimuli are presented together, then there is some evidence that the total prediction is shared between them, possibly unequally, as if one (in this case the sound) is more salient than the other. Minimising $E(\mathbf{w})$ only requires that

$$w_l + w_s = r \tag{10.7}$$

and so altering the learning rule to give each stimulus its own associability:

$$w_l \rightarrow w_l + \epsilon_l (r - v) u_l \tag{10.8}$$

$$w_s \rightarrow w_s + \epsilon_s (r - v) u_s \tag{10.9}$$

and setting $\epsilon_l < \epsilon_s$ will lead to the observed result.

Note that the linear prediction model in figure 10.4 may be too simple. In particular, various other models consider interactions between the representations or associabilities of the stimuli, making dependent the numerical (and no longer binary) values of u_l and u_s or ϵ_l and ϵ_s. Further, on the basis of other, more complicated experiments, the animals can be shown to know that the light and sound have been presented simultaneously. Generally, animals learn much more about the causal texture of their environment (Dickinson, 1980) than just the relationships between stimuli and rewards and appropriate actions, and the ways this knowledge affects their learning and behavior are under active investigation.

Blocking, the final example in table 10.3, is the paradigm that led to the suggestion of the delta rule in the first place. If as a result of the first set

Name	Set 1	Set 2	Test
Inhibitory		$\left\{\begin{array}{c} l \to r \\ l+s \to \cdot \end{array}\right\}$	$s \overset{?}{\to} -$
Overshadowing		$l+s \to r$	$\left\{\begin{array}{c} l \overset{?}{\to} r^- \\ s \overset{?}{\to} r^+ \end{array}\right\}$
Blocking	$l \to r$	$l+s \to r$	$s \overset{?}{\to} \cdot$

Table 10.3: Conditioning paradigms: inhibitory conditioning, overshadowing and blocking. After learning in inhibitory conditioning, it can be shown by a variety of means that the sound predicts the *absence* of reward ($s \overset{?}{\to} -$). In overshadowing, association with the reward can be divided unequally between the sound and the light, indicated by r^- and r^+.

of trials, $w_l = r$, then during the second set, the prediction error $r - v = 0$ and so no learning ensues to either the light or the sound. In particular no association will be made between the sound and the reward. This is a path dependent effect and not just an extreme form of overshadowing – swap the light and the sound, and the light will duly attract no conditioning.

The final paradigm we consider is secondary conditioning, which suggests that the conditioned stimuli like sounds and lights can directly control the systems that report on rewards themselves. Table 10.4 shows that if the light is conditioned to the reward in one set of trials, then it can be used to train positive predictions to the sound in another set of trials. Ostensibly, this might seem like a case for extinction or inhibitory conditioning of the light, or both. Extinction should happen since the light is being shown without the reward. Indeed the association between the light and the reward does extinguish during the course of the second set of trials, but not before the sound has had the chance to demonstrate a positive association.

On the other hand, inhibitory conditioning might seem to be the natural result of a secondary conditioning experiment, since the sound should come to predict the absence of reward that used to be delivered with the light. From the perspective of the delta rule, the prediction error $r - v$ is *negative* in the second set of trials. For, if $w_s = 0$ at the outset, then $v = w_l$ for the light which has been conditioned by the first set of trials to predict the presence of reward, but $r = 0$, since no reward is delivered in the second set. The fact that the delta rule arranges for the wrong direction of learning suggests that the whole goal of minimising $E(\mathbf{w})$ may be flawed. The replacement will lead to an explanation not only of secondary conditioning, but also of the data in figure 10.2 on the firing of the dopamine cells during conditioning, and also one way to solve the maze task of figure 10.1, with its problems of temporally extended prediction and control.

Name	Set 1	Set 2	Test
Secondary	$l \to r$	$s \to l$	$s \overset{?}{\to} r$

Table 10.4: Conditioning paradigm: secondary conditioning. Only a limited number of trials is used in set 2.

Time-Based Conditioning

A key difference between inhibitory and secondary conditioning turns out to lie in the precise time course of the presentation of the stimuli. In the same way that the bell tolled for Pavlov's dogs *before* the meat was given, in many circumstances, it is important to present the conditioned stimuli a little *before* the reward to get good conditioning to the reward. The optimal temporal separation differs for different cases of conditioning. In conventional secondary conditioning experiments, the sound only acquires a positive association with the reward if it *precedes* the light in set 2 in the same way that the light precedes the reward in set 1.

Incorporating the detailed time-course of learning into a model requires a number of changes. First, a new formal definition of the prediction task is required which takes account of time within a trial. Second, if the light is shown briefly so that it is not present when the reward is delivered, there might appear to be no conditioned stimulus present at all by the time the reward is given. How could there ever be any learning? Third, there is evidence that animals know how long it takes for the reward will be delivered after the light – there must be some substrate for this prediction.

The formal nature of the temporally sensitive prediction task is that the presentation of the light at one time t during the trial has to predict the delivery of a reward at a later time $t+\tau$ in the same trial. Further, since each single trial is separated into its detailed time-course, it also becomes possible for reward to be delivered at more than one time in a trial. So, instead of using $v = \mathbf{w} \cdot \mathbf{u}$ to predict r, a natural goal in the time dependent case is to use the stimuli present at time t, $\mathbf{u}(t)$ to predict the *sum* total reward to be delivered in the trial *after* time t, *ie* to use

$$v(\mathbf{u}(t)) \equiv \mathbf{w} \cdot \mathbf{u}(t) \tag{10.10}$$

to predict

$$\left\langle \sum_{\tau \geq 0} r(t+\tau) \right\rangle , \tag{10.11}$$

averaging over the stochastic delivery of rewards and possibly also the stochastic presentation of stimuli within a trial. We ignore these stochastic aspects for the present, dropping the averages and treating the rewards as if they are deterministic. The issue of stochasticity will be revisited in section 10.4, where the rules we are about to describe are discussed in substantially more detail.

We write $v(t)$ for $v(\mathbf{u}(t))$ where not confusing. Arranging for $v(t)$ to predict the sum future reward would seem to require only a simple alteration to the delta rule of equation 10.6:

$$\mathbf{w} \rightarrow \mathbf{w} + \epsilon \left(\sum_{\tau \geq 0} r(t+\tau) - v(t) \right) \mathbf{u}(t) \tag{10.12}$$

using the full sum of rewards in the future of a trial rather than just r. If the sum reward in equation 10.11 were known at time t, this would indeed be easy. However, although $r(t)$ is known at the same time as $\mathbf{u}(t)$, the succeeding $r(t+1), \ldots$ have yet to be experienced, making it hard to calculate the prediction error in equation 10.12, which we write in an equivalent form as

$$r(t) + \sum_{\tau \geq 1} r(t+\tau) - v(t) \ , \tag{10.13}$$

If the predictions of future reinforcement were correct, then the prediction at time $t+1$ would satisfy

$$v(t+1) = \sum_{\tau \geq 1} r(t+\tau) \tag{10.14}$$

The *temporal difference* model of prediction takes the simple step of using the prediction $v(t+1)$ *in place of* $\sum_{\tau \geq 1} r(t+\tau)$ and so approximates the overall prediction error of equation 10.13 by

$$\delta(t) = r(t) + v(t+1) - v(t) \ . \tag{10.15}$$

This term $\delta(t)$ is called the temporal difference, since it involves the difference between two successive estimates $v(t+1) - v(t)$. This prediction error can then be used exactly as in the standard delta rule:

$$\mathbf{w} \rightarrow \mathbf{w} + \epsilon \delta(t) \mathbf{u}(t) \tag{10.16}$$

for each timestep t in the trial. The overall rule is called the temporal difference learning rule (Sutton, 1988; Sutton & Barto 1989). There is an extensive body of theory showing circumstances under which this rule will converge appropriately on the optimal predictions, including cases for which the delivery of reward is stochastic rather than deterministic.

In one model, the dopamine cells in the VTA whose activity is shown in figure 10.2 report the prediction error for reward $\delta(t)$ (Montague *et al*, 1996). This signal controls the learning of predictions and (as we see later) learning of actions. Crudely, before learning, the delivery of the reward is unexpected, and so the prediction error is positive at the time of the reward. After learning, the monkey knows to expect the reward on the basis of the cue, and so there is no prediction error at the time of the reward. However, before the delivery of the light, it does not know it is about to get a reward. After the delivery of the light it knows it is, and so there

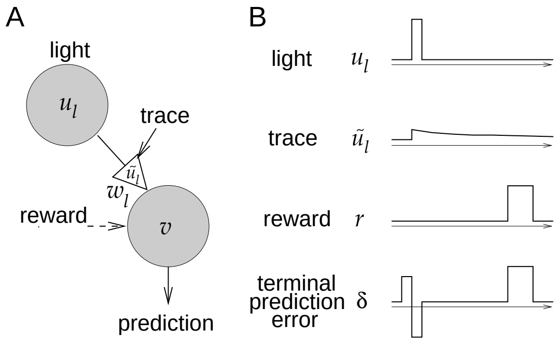

Figure 10.5: The trace rule. A) The trace of the past occurrence of the light $\tilde{u}_l$ only affects synaptic plasticity and not the actual predictions. B) The effect of using traces with the TD prediction error. The trace is shown in the second graph; the lowest graph shows the terminal prediction error, revealing that the model has no expectation as to the time of delivery of the reward.

is a positive prediction error associated with the light. These traces suggest a natural account of both secondary conditioning and blocking. Early on, the dopamine cells fire to the reward, and so support learning about the reward by the light, which *precedes* it. Later, the dopamine cells fire to the light and not the reward, and so if a second stimulus such as a sound were to precede the light, then the sound should become positively rather than negatively conditioned. Conversely, if the sound were presented at the same time as the light, then since there is no non-zero prediction error later in the trial (as the predictions are correct), there would be no learning to the sound.

The second problem posed by secondary conditioning is that no conditioned stimuli need be present when the reward is actually delivered – this is clear by design in the monkey experiment in figure 10.2. An important class of models to address this is based on stimulus traces. For these models, the *predictions* $v(t)$ are based only on the immediate presence or absence of stimuli $\mathbf{u}(t)$ at particular times. However, once a stimulus has been delivered, it remains eligible for learning for some non-negligible future time. The system retains a *trace* of the fact that the light was on, and through this trace, is able to associate future prediction errors $\delta(t+t')$ for $t' > 0$ with the light. For instance, in one trace model shown in figure 10.5A, each pre-synaptic unit maintains an eligibility trace. For the light, this eligibility trace $\tilde{u}_l(t)$ develops according to a low pass filter:

$$\tilde{u}_l(t) = \tilde{u}_l(t-1) + (1-\lambda)(u_l(t) - \tilde{u}_l(t-1)), \qquad (10.17)$$

whose decay is determined by $0 \leq \lambda \leq 1$. The smaller λ, the faster the decay. The equivalent of the temporal difference rule in equation 10.16 becomes:

$$\mathbf{w} \rightarrow \mathbf{w} + \epsilon\delta(t)\tilde{\mathbf{u}}(t) \ . \qquad (10.18)$$

Rules involving traces have been shown to model successfully a variety of conditioning data. However, there is a theoretical and empirical problem about timing raised in the experiment in figure 10.2 that they cannot address. Imagine that the model is correct, *ie* that the dopamine cells really report the prediction error $\delta(t)$ of equation 10.15. Consider what happens at the time step that the reward is delivered (call this time t^*). Since there is apparently no stimulus present, $v(t^*+1)$ and $v(t^*)$ should both be 0, and so $\delta(t^*)=r(t^*)=r$. This would suggest that the dopamine cells would still fire on the presentation of the reward, which, by figure 10.2, is incorrect. Stimulus traces enable learning to accrue to stimuli that are not immediately present, but do not support the prediction of exactly when in a trial the reward will appear.

Figure 10.5B shows the consequences of using the trace rule. It depicts the time courses of the light and its stimulus trace, the reward, and the prediction error $\delta(t)$ once the net change to the weight in equation 10.18 during the course of a trial is 0. We can see that although the prediction error does indeed become positive at the time the light is presented, it goes negative straight afterwards, of which there is no hint in figure 10.2. Furthermore, the firing at the time of the reward is undiminished. What is going wrong is that although the animal is apparently expecting the delivery of reward throughout the time between the presentation of the light and the delivery of the reward, there is no mechanism in the model for maintaining this prediction across the temporal gap. Figure 10.6 gives another hint as to the problem. This shows the activity of dopamine cells whilst monkeys were learning which of two levers to press in a somewhat similar task to the original reaction-time task. Usually the animals would make the correct choice of button, giving the trace on the left (the activity at the time of the reward has yet to be learned away, as in figure 10.2). In the occasional trial in which they picked the wrong button, there is a *decrease* in the activity of the cells just when the activity consequent on the reward would have been expected. This is more evidence that the monkeys know *when* reward will be provided.

To be able to expect the reward a certain time after the stimulus, the animal must be able to keep time between the two. A simple way of doing this is to represent the light not by $u_l(t)$, but by a whole set of representations $u_l^\tau(t)$ indexed by a variable τ that counts how long it is since the light came on in a trial, where

$$u_l^\tau(t) = \begin{cases} 1 & \text{if } u_l(t-\tau) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{10.19}$$

These are shown symbolically in the model in the upper part of figure 10.7. There are other representations that provide similar information, but have slightly different consequences for the course of learning. The sound is similarly represented by a whole family of representations $u_s^\tau(t)$.

Each member of both families of predictive stimuli has its own weight w_l^τ

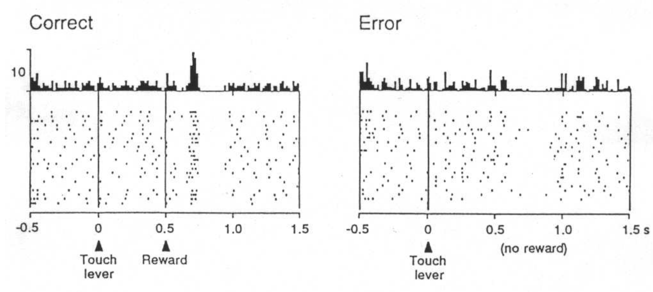

Figure 10.6: Error activity. Left and right panels show the activity of the same dopamine neuron during the learning of a particular task involving a choice between two levers. The left panel shows the activity following delivery of the reward when the monkey makes the correct choice – when learning is complete, for most cells, this activity then disappears as in figure 10.2. The right panel shows the activity following an incorrect response. The basal firing rate of dopamine cells is rather low – but not so low that the inhibition of the cells at just the time the reward would have been expected is not evident. From Schultz, Apicella & Ljungberg (1993).

and w_s^τ, the total prediction is:

$$v(t) = \sum_\tau w_l^\tau u_l^\tau(t) + w_s^\tau u_s^\tau(t) \tag{10.20}$$

and the learning rule, based on equation 10.16, is

$$\mathbf{w}^\tau \rightarrow \mathbf{w}^\tau + \epsilon\delta(t)\mathbf{u}^\tau(t) \tag{10.21}$$

This rule can also be combined with an eligibility trace and different associabilities.

The lower part of figure 10.7 shows what happens during learning in an analogue of the experiment in figure 10.2. These traces should be contrasted with those for the trace model shown in figure 10.5B. The complete set of stimuli representing the light allow the model to count the time between the light and the reward, and it is apparent from the 'after' trace of $v(t)$ that the prediction remains high throughout this period. Even though the prediction remains high, one would not expect the prediction error $\delta(t)$ to be persistently active. Indeed experiments on the dopamine cells of monkeys designed to test exactly this, showed that there is generally not persistent activity during such delay periods. Further, both the two unreasonable features of $\delta(t)$ in the trace model, the negative blip just after the light is extinguished and the continued presence of firing on presentation of the reward, are gone. Note that the prediction difference term $v(t+1) - v(t)$ is negative at the time the reward is delivered. This is the model's account of the inhibition apparent in figure 10.6 when reward is denied on a trial.

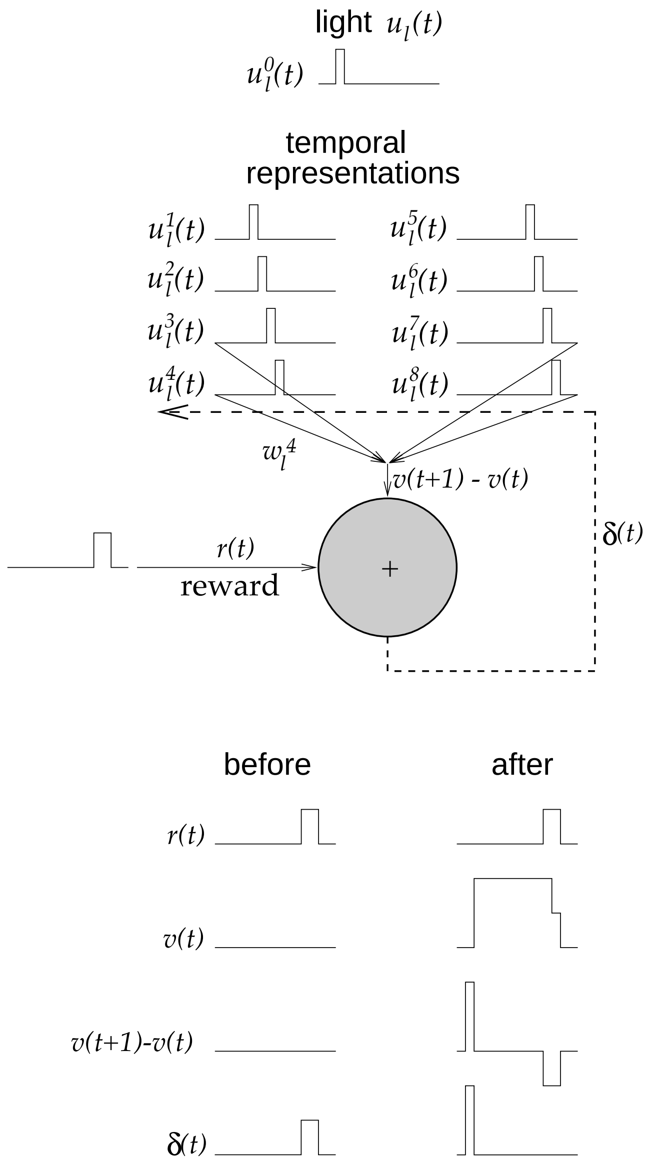

Figure 10.7: Temporal difference model for conditioning. The top part shows the set of stimuli that jointly represent the light, each with its own prediction weight. The lower part shows traces for the various signals in the model before and after learning. The reward lasts for two time-steps, which is why the prediction inbetween the light and the reward is two units rather than just one.

Alternative Models

The delta and temporal difference rules are notably successful at accounting for large bodies of conditioning data, in making a link to well-founded computational notions of prediction, and to understanding the behavior of critical neural signals such as the activity of the dopamine cells. The success of the delta rule led to a wide variety of behavioral tests of its adequacy, in many of which, it, and also the temporal difference rule, have been found wanting. For instance, extinction (in table 10.2) is not as simple as setting the prediction weight w_l to 0 – there is evidence that animals do not forget that the light was once associated with the reward. Equally, although the delta rule can account for the fact in partial reinforcement (table 10.2) that the asymptotic prediction w_l should be less than r, it cannot account for the further fact that extinction is far slower following partial than full reinforcement. Also, unlike standard Pavlovian conditioned stimuli, conditioned inhibitors (see table 10.3) do not extinguish just by being presented by themselves.

Many of these findings can be accounted for at a computational level by making different assumptions about either the goal or the statistical model underlying prediction, and at an algorithmic level by making different assumptions about the mechanisms for satisfying the resulting prediction goals. Rather than discuss alternative models in detail, we will consider some particular computational and algorithmic suggestions.

Rate Estimation

The computational basis of the delta rule model is prediction of the expected value of a reward in a trial. If rewards are modeled as being just present or absent, the expected value of a reward is just its *probability* of occurring. Gallistel and Gibbon (1998) have suggested a computational model that is based on the different principle of predicting the *rate* of delivery of reward. Although this may seem like a small change – a rate is just a probability per unit time in a Poisson process – it actually has rather far reaching consequences because of the way that rates for the CSs and rates for the conditioning context seem to be combined.

Figure 10.8 shows the key results for which this model is intended to account. It is based on studies into the acquisition and extinction of a classically conditioned peck response of a pigeon to a light that is associated with the delivery of food in a hopper. This is also known as *autoshaping*. Various groups of birds were tested, probing the effects of the two variables shown in figure 10.8A. One variable is the I/T ratio, which is the inter-stimulus interval divided by the time in a trial for which the light is presented. The larger the I/T ratio, the rarer the light within a trial. The other variable is the training schedule S, which is the fractional number of food deliveries per trial – some of the birds were only partially reinforced.

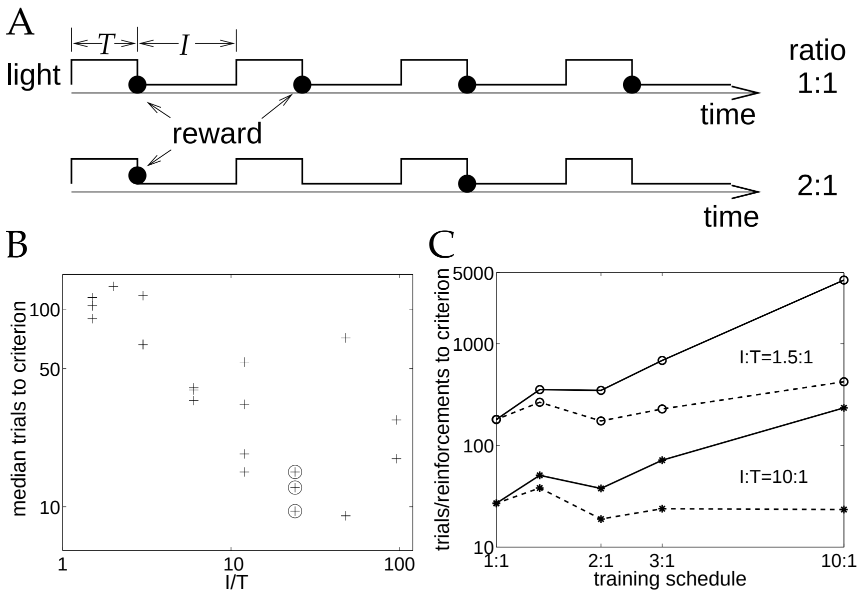

Figure 10.8: Invariances in conditioning. A) Experimental paradigms for autoshaping. The upper figure shows a case in which food follows the termination of every single conditioned stimulus (light); the lower a case where the training schedule is 2:1. T is the time during the trial for which the light is presented; I is the inter-trial interval. B) Log-log plot of median number of trials to acquisition criterion as a function of I/T for $S=1$. C) Log-log acquisition curves for $I/T = 1.5$ (cirles) and $I/T = 10$ (stars) as a function of the training schedule S. Solid curves show the median number of *trials* to reach an acquisition criterion; dashed curves show the median number of *reinforcements* to criterion. Data in (B) from Gibbon *et al* (1977); data in (C) from Gibbon *et al* (1980); both as analysed by Gallistel & Gibbon (1998).

During acquisition, the number of trials necessary before the birds would peck at least once on three out of four successive trials was recorded (this is the acquisition *criterion.* For the cases of partial reinforcement, these data are also used to calculate the number of delivered or omitted reinforcements until acquisition or extinction.

Figure 10.8B shows how long acquisition takes as a function of the I/T ratio, in the case that reward is provided on every trial ($S=1$). Different points on the graph come from dramatically different values of I and T. For instance, from top to bottom, the three circled values have $(I, T) = (96, 4)$, $(768, 32)$ and $(384, 16)$ seconds. This graph suggests that I/T is the critical variable governing the speed of acquisition, and that the rarer the light in a trial (*ie* the larger I/T), the faster is acquisition. Once learning is complete, though, it turns out that the ultimate rate of pecking of the birds during the light is mostly independent of I, and only depends (somewhat weakly) on T.

Figure 10.8C shows how the speed of acquisition depends on the partiality of reinforcement (S). The solid lines show the median number of *trials* to acquisition as a function of S for two values of the I/T ratio. The effect from figure 10.8B is apparent – for all values of S, acquisition is slower for smaller I/T. Furthermore, the rarer are the rewards (the thinner the training schedule), the slower is learning. This makes sense since fewer rewards are received in the same number of trials. More interestingly, if one measures the time for acquisition in terms of the number of *reinforcements* delivered, normalising away this effect, then the acquisition curves (dashed lines) are roughly flat, showing that S has no effect.

These rather precise relationships pose a challenge to standard accounts of conditioning. Trial-based learning rules such as the Rescorla-Wagner rule are completely insensitive to the I/T ratio, but are strongly affected by partial reinforcement (since un-learning happens during each non-presentation of the reward). The data show exactly the reverse.

Gallistel & Gibbon explain the results by suggesting that the birds are estimating *rates* of delivery of reinforcement, one for the rate of reinforcement apportioned to the light ($\hat{\lambda}_{CS}$) and the other for the rate apportioned to the conditioning context ($\hat{\lambda}_b$). The model further suggests that they should choose to respond if:

$$\frac{\hat{\lambda}_{CS} + \hat{\lambda}_b}{\hat{\lambda}_b} > \zeta \tag{10.22}$$

where ζ is a threshold. The idea is that if the total expected rate of delivery during the time the light is present (including the background rate) is a sufficient multiple of the expected rate when the light is not present, then the expectation of reward is sufficient for the pigeon to deem it worth responding. Equivalently, if

$$\hat{\lambda}_{CS} > \zeta' \hat{\lambda}_b \tag{10.23}$$

where $\zeta' = \zeta - 1$, then the animals should respond. The model takes for granted the ability of the pigeon to count, accumulate, and manipulate intervals of *time* and also *numbers* of stimuli. We will describe a restricted portion of the model that is designed to account for the data in figure 10.8.

Although Gallistel & Gibbon modeled the pigeons as frequentist statisticians, it is a bit easier to conceive of them as Bayesians. If reinforcements are modeled as coming from a constant rate Poisson process, then the probability of receiving n_{CS} rewards during the time t_{CS} that the light has been present is:

$$P[n_{\mathrm{CS}}|t_{\mathrm{CS}};\lambda_{CS}] \propto e^{-(\lambda_{CS} t_{\mathrm{CS}})}\,(\lambda_{CS})^{n_{\mathrm{CS}}} \tag{10.24}$$

So, if the bird's prior over λ_{CS} is essentially flat,

$$P[\lambda_{CS}|n_{\mathrm{CS}};t_{\mathrm{CS}}] = e^{-(\lambda_{CS} t_{\mathrm{CS}})}\,(\lambda_{CS})^{n_{\mathrm{CS}}}\,\frac{(t_{\mathrm{CS}})^{n_{\mathrm{CS}}+1}}{n_{\mathrm{CS}}!} \tag{10.25}$$

is a gamma distribution, whose mean is

$$\hat{\lambda}_{CS} = \frac{n_{\rm CS} + 1}{t_{\rm CS}}. \tag{10.26}$$

The background rate is estimated similarly, except that, in the simple conditions of figure 10.8A, no food is delivered during the background. Also, the background is presented for the entire length of the conditioning experiment – the food deliveries that are received whilst the light and the background are both present are apportioned to the light. By similar reasoning, this implies that

$$\hat{\lambda}_b = \frac{1}{t_{\rm b}}. \tag{10.27}$$

Therefore, the condition of equation 10.23 translates into

$$\frac{n_{\rm CS} + 1}{t_{\rm CS}} > \zeta' \frac{1}{t_{\rm b}} \qquad \text{or} \qquad n_{\rm CS} > \zeta' \frac{t_{\rm CS}}{t_{\rm b}} - 1 \tag{10.28}$$

Since $t_{\rm CS}/t_{\rm b} = T/(I+T) = 1/(1+I/T)$, this means that the number of reinforcements required is almost inversely proportional to the I/T ratio, as roughly observed in the data. This also explains the acquisition portion of the partial reinforcement effect. Consider the case that the training schedule S is $2:1$, so that reinforced and non-reinforced trials are alternated. If I and T are summed between rewards, then they are both effectively doubled, and so I/T, and hence the required number of reinforcements to acquisition, is left constant.

There are some puzzling features about the relationship between the rate of acquisition and I/T. In particular, acquisition is extremely slow compared with a statistically ideal observer, who would calculate the probability $P[\lambda_l > \zeta' \lambda_b | n_l, n_b; t_l; t_b]$ rather than comparing the ratios of the posterior means $\hat{\lambda}_{CS}/\hat{\lambda}_b$. Nevertheless, it poses an important theoretical challenge.

Associabilities and Opponent Processes

Gallistel and Gibbon suggest an alternative computation in the sense of a different intent for prediction. There are rather more suggestions at an algorithmic level for avoiding the flaws of the delta and temporal difference learning rules. Two of these suggestions are particularly important. One is that many of even the basic phenomena of conditioning should be ascribed to varying associabilities or degrees of representation of the stimuli; the other is that rather than only considering a single prediction error for reward, it is essential to consider the interactions among multiple systems reporting information about different drives and about rewards and punishments. In many models, these systems can be considered in pairs, such as reward and punishment, each member of which opposes the other. Such antagonistic pairs are sometimes called *opponent systems.*

One model (amongst many) that includes sophisticated versions of both these effects is due to Grossberg (see 1988). In it, there is a dynamic competition (called an attentional competition) between different conditioned stimuli and between conditioned and unconditioned stimuli, on the basis of their intrinsic and acquired saliencies. Stimuli that lose the competition are not available to make predictions or undergo learning – however, if something unexpected happens, then a stimulus that previously lost the competition can be revived in order to explain the novel findings. Consider the phenomenon of blocking, described in table 10.3. For the temporal difference model, prediction *errors*, as reported by $\delta(t)$ control the course of learning. In blocking, after the first set of trials has established the light as a good predictor of the reward, there are no prediction errors, and so no opportunity for learning to the sound. For Grossberg's model, prediction errors and novelty exert control over the associabilities of stimuli, and, in particular, the relationship between the light and reward established in the first set of trials means that the light beats out the sound in the competition to be represented when the second set of trials starts. Since there is no prediction error in the second set of trials, the sound never wins the competition in a sustained manner, and so attracts no learning.

Figure 10.9 shows one of the versions of opponency that this model has employed – a mechanism called a gated dipole. The gating refers to the square synapses on the connections from cell a_1 to a_3 and cell a_2 to a_4 which are treated qualitatively as if they show effects of transmitter depletion – *ie* the more active the total input is, the more response there is at the beginning, but the less neurotransmitter is left at the end. These gating synapses are embedded in a dipole architecture in which there are positive (ON) and negative (OFF) channels that inhibit each other. This means that whereas the ON-channel and OFF-channel have similar responses to the sustained input I at the beginning of the traces, and therefore cancel each other out, the ON-channel's response is significantly weaker at the end, allowing for a transient OFF-response (called a rebound) at the end of the pulse in the transient input J. Such transients may be important, for instance, in reporting the aversive characteristic of the *termination* of the provision of an extended reward.

An example of an effect produced by the dipole model is active rather than passive extinction of predictors of reward. Rather than just setting the weight from the stimulus unit u_l to the ON-channel to 0, one can set the weight from u_l to the OFF-channel to be large. The experimentally wayward course of extinction (for instance animals can sometimes 'forget' that a stimulus-reward association has been subject to extinction) may be explained by properties of this active process. Further, it points out the over-simplicity of the model of conditioned inhibition used above that has $w_l < 0$. In fact, the issue of exactly what a conditioned inhibitor predicts is rather tricky – whether it is the absence of reward, the absence of a particular reward (such as a type of food), the presence of a punishment, or only the negation of the predictions of reward made by other stimuli.

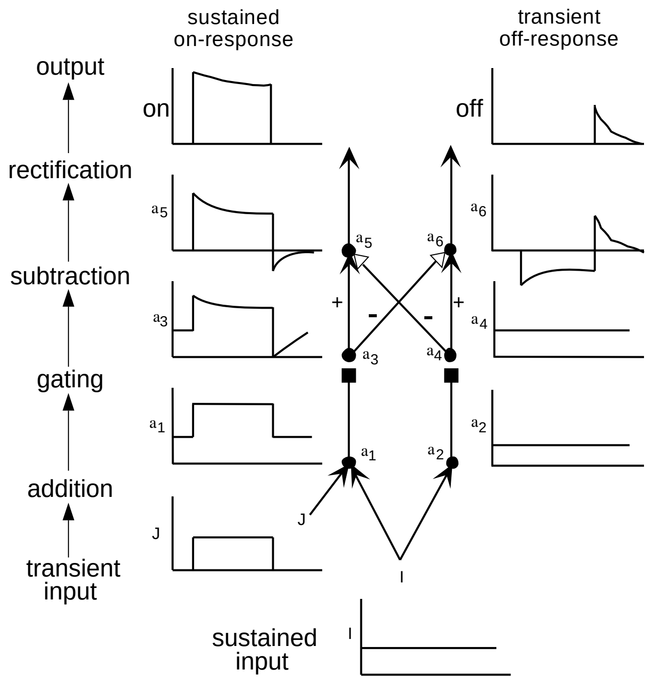

Figure 10.9: A feedforward gated dipole. Left hand units comprise the ON channel; right hand units the OFF channel. The connections are linear with the signs shown except for the square synapses which are the depleting transmitter gates. The layers in the network perform the operations shown on the left. The graphs show the consequence of applying a phasic signal J to the ON channel, and equal tonic signals I to the ON and OFF channels. When J is present, the higher input in the ON channel leads to a sustained output. When J ends, the depletion of the transmitter in synapse between a_1 and a_3 makes the response of a_3 to the remaining tonic input transiently depressed and so leads to a transient OFF response. Grossberg's current models use recurrent versions of this gated dipole. After Grossberg (1988).

It is impossible to do full justice to such a complex model with qualitative descriptions. However, the circular interaction between the attentional competition between stimuli, the learning that then accrues to them, and then the attention that follows from that learning, is responsible for a number of interesting effects. The interactions between ON and OFF channels has direct impact on puzzling effects in the malfunctions of these reinforcement systems in diseases and addiction.

10.3 Action Choice

The second component of solving the full maze problem of figure 10.1 comes from the instrumental conditioning task of learning to choose the appropriate action given direct rewards. There is again extensive theoretical and experimental work on solving this problem. We consider just two simple methods, one direct and the other indirect. The relationship with psychological data is much more distant in this case, although all the models follow the basic tenet that actions that lead to rewards should become more likely to be selected. Further, many were originally suggested as abstractions of results in instrumental conditioning.

The Direct Actor

Consider the choice of direction at point **s** in the case that a unit reward is delivered for going right (action R) and nothing is delivered for going left (action L). A simple learning strategy is to maintain probabilities $p[R] = 1 - p[L]$ and update them in the light of the delivery of reward. One standard rule is to increase

$$p[R] \to p[R] + \eta(1 - p[R]) \qquad \text{given reward on going R} \tag{10.29}$$

where $0 < \eta < 1$ is a learning rate. The factor $1-p[R]$ in equation 10.29 makes learning slow down as $p[R] \to 1$, and keeps $p[R] \le 1$.

Given the choice of going left at **s**, and consequently receiving no reward, an additional possibility is to decrease $p[L]$ according to:

$$p[L] \to p[L] + \eta(0 - p[L]) \qquad \text{given no reward on going L} \tag{10.30}$$

Since $p[L] + p[R] = 1$, we can write this equivalently as

$$p[R] \to p[R] + \eta(1 - p[R]) \qquad \text{given no reward on going L} \tag{10.31}$$

which is just the same as the change in equation 10.29. Since $p[R]$ is actually the probability of choosing to go right, the average change to $p[R]$ is

$$\eta\left(p[R](1 - p[R]) + p[L](1 - p[R])\right) = \eta(1 - p[R]) \tag{10.32}$$

and so $p[R]$ tends to 1 exponentially quickly. The reason not to set $p[R]$ to 1 as soon as the first reward is received for going right is that rewards might be stochastic and going left could ultimately be more lucrative. This task is an example of a two-armed bandit problem (Berry & Fristedt, 1985), where pulling an arm of the slot machine is like selecting one of the two directions in the maze. There is a large body of theory investigating the capacities of algorithms like this (which are called linear algorithms) for solving two armed bandit problems, and in investigating the optimal trade-off between exploration and exploitation from a Bayesian perspective.

We need two generalisations of this. The first is to the case that there are real valued and stochastic rewards and also many actions, chosen from a set $\mathcal{A}$, rather than just the two. One convenient way to allow for multiple actions is to use a set of action values s^a and the *softmax* probability distribution over actions such that

$$p[a] = \frac{e^{s^a}}{\sum_{a' \in \mathcal{A}} e^{s^{a'}}} \tag{10.33}$$

We write $\mathbf{S}$ as the set of all action values $\{s^a\}$. Consider the case that the rewards r are real-valued and stochastic rewards r, depending on the action. We write $\langle r \rangle^a$ for the mean reward for taking action a. Then, one appropriate objective for the action vectors $\mathbf{S}$ is to try to maximise the expected reward:

$$E(\mathbf{S}) = \sum_a p[a] \langle r \rangle^a \tag{10.34}$$

Given a trial on which action a is selected, giving reward r, consider a learning rule of the form

$$s^b \quad \rightarrow \quad s^b + \eta(r - \bar{r}) \nabla_{s^b} \log p[a] \tag{10.35}$$

$$\quad = \quad s^b + \eta(r - \bar{r})(\delta_{ab} - p[b]) \tag{10.36}$$

where $\bar{r}$ is any number that does not depend on the action or the particular reward received and δ_{ab} is the Kronecker delta function. Since the expected value $\sum_a p[a]\delta_{ab} = p[b]$, if $\bar{r} = E(\mathbf{S})$, this would actually be a covariance rule, as studied in chapter 8. Since

$$\nabla_{s^b} \log p[a] = \frac{\nabla_{s^b} p[a]}{p[a]} \, , \tag{10.37}$$

the average change to s^b based on equation 10.35 is proportional to

$$\sum_a p[a](\langle r \rangle^a - \bar{r}) \frac{\nabla_{s^b} p[a]}{p[a]} \quad = \quad \sum_a \nabla_{s^b} p[a] \langle r \rangle^a - \bar{r} \sum_a \nabla_{s^b} p[a] \tag{10.38}$$

$$\quad = \quad \nabla_{s^b} E(\mathbf{S}) - \bar{r} \nabla_{s^b} \sum_a p[a] \tag{10.39}$$

$$\quad = \quad \nabla_{s^b} E(\mathbf{S}) \tag{10.40}$$

since $\sum_a p[a] = 1$ is constant (Williams, 1992). The average is taken over the stochasticity of both the rewards *and* the choice of actions. This result means that the learning rule in equation 10.35 performs stochastic gradient ascent on the expected reward $E(\mathbf{S})$, irrespective of the value of $\bar{r}$, and so at least locally optimises the action. The only control over the balance between exploration and exploitation is the learning rate η – if the initial action values are all the same, then the smaller η, the longer the exploration; the larger η, the quicker the (possibly suboptimal) exploitation.

Stochastic gradient ascent is not a panacea for all the ills of optimisation – in particular, it can be painfully slow.

Another way of looking at the learning rule is that for the action a actually chosen, s^b increases if $r > \bar{r}$ and decreases if $r < \bar{r}$. The quantity $\bar{r}$ therefore acts as a form of baseline against which the rewards are judged. Paradoxically, the learning rule performs stochastic gradient ascent whatever the value of $\bar{r}$, because even if $\bar{r}$ is much smaller than all the rewards, say, then s^{a^*} for the best action a^* will increase faster than s^a for all the other actions. A natural value for $\bar{r}$ is just $E(\mathbf{S})$.

The final generalisation is to the case of multiple locations or states. One could just have different sets of action values, one for each state. A slightly more general alternative, which fits in better with network models such as that in figure 10.4 is to have the state be signalled by a vector $\mathbf{u}$. In this case, a modest modification is to replace the action numbers s^a with action vectors $\mathbf{s}^a$, and replace equation 10.33 with

$$p_{\mathbf{u}}[a] = \frac{e^{\mathbf{s}^a \cdot \mathbf{u}}}{\sum_{a' \in \mathcal{A}} e^{\mathbf{s}^{a'} \cdot \mathbf{u}}} \tag{10.41}$$

This collection $\mathbf{p}$ of distributions over actions $p_{\mathbf{u}}[a]$, one for each state $\mathbf{u}$ is called a *policy.* The resulting action choice network is shown in figure 10.10. Given this definition, the learning rule for the action vectors is almost unchanged from equations 10.35 and 10.36. Based on choosing action a at state $\mathbf{u}$ and getting reward r we use:

$$\mathbf{s}^b \quad \rightarrow \quad \mathbf{s}^b + \eta(r - \bar{r})\nabla_{\mathbf{s}^b} \log p_{\mathbf{u}}[a] \tag{10.42}$$

$$= \quad \mathbf{s}^b + \eta(r - \bar{r})(\delta_{ab} - p_{\mathbf{u}}[b])\mathbf{u} \,. \tag{10.43}$$

If $\langle r \rangle^a_{\mathbf{u}}$ is the average reward for performing action a at state $\mathbf{u}$, then the learning rule in equation 10.42 performs stochastic gradient ascent in the new mean reward

$$E(\mathbf{S}) = \sum_{\mathbf{u}} p[\mathbf{u}] \sum_{a} p_{\mathbf{u}}[a] \langle r \rangle^a_{\mathbf{u}} \tag{10.44}$$

which is averaged also over the distribution $p[\mathbf{u}]$ of states. More complicated parameterised schemes than equation 10.41 for representing the policy lead, via equation 10.42, to more complicated stochastic gradient ascent rules for the parameters. Questions of the representation of state are not the focus of this chapter.

The Indirect Actor

An alternative method for learning the appropriate actions is to learn the expected values of performing action a at state $\mathbf{u}$ and then explicitly choose

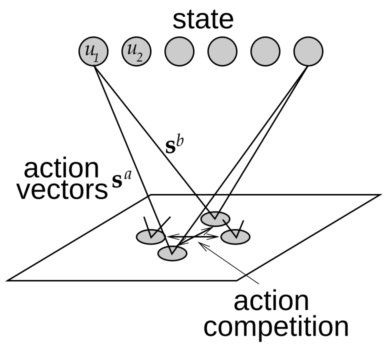

Figure 10.10: Policy network. Action vectors $\mathbf{s}^a, \mathbf{s}^b, \ldots$ are weights from units $\mathbf{u}$ representing the state to units representing the actions, which compete via the denominator in the softmax equation 10.41.

actions to optimise the choice. As for classical conditioning, the delta rule:

$$Q_{\mathbf{u}}(a) \to Q_{\mathbf{u}}(a) + \epsilon(r - Q_{\mathbf{u}}(a)) \tag{10.45}$$

can be used to make the estimates $Q_{\mathbf{u}}(a)$ close to the true average values $\langle r \rangle_{\mathbf{u}}^a$. The only difference from standard classical conditioning is that $Q_{\mathbf{u}}(a)$ depends on the choice of action a.

A convenient way to choose an action at state $\mathbf{u}$ is again to use the softmax probability distribution

$$p_{\mathbf{u}}[a] = \frac{e^{\beta Q_{\mathbf{u}}(a)}}{\sum_{a' \in \mathcal{A}} e^{\beta Q_{\mathbf{u}}(a')}} \tag{10.46}$$

where $\beta > 0$ is used to control the balance between exploration and exploitation. The larger β, the more likely an action a^* for which

$$Q_{\mathbf{u}}(a^*) = \max_a Q_{\mathbf{u}}(a) \tag{10.47}$$

is to be chosen, *ie* the more the balance is tipped in favor of exploitation.

Despite the superficial similarity, there is a critical difference between the two softmax terms in equations 10.41 and 10.46. Equation 10.46, based on the values $Q_{\mathbf{u}}(a)$ inevitably performs a form of probability matching – if the estimated values of two actions are very similar, then their probabilities of selection will be very similar too, unless β is very large. Conversely, in equation 10.41, there is no immediate relationship between $\mathbf{s}^a \cdot \mathbf{u}$ and the expected rewards $\langle r \rangle_{\mathbf{u}}^a$. Rather if an optimal action is a^*:

$$\langle r \rangle_{\mathbf{u}}^{a^*} = \max_a \langle r \rangle_{\mathbf{u}}^a \tag{10.48}$$

then $\mathbf{s}^{a^*} \cdot \mathbf{u} - \mathbf{s}^a \cdot \mathbf{u}$ should tend to increase for all suboptimal a, making $p_{\mathbf{u}}[a] \to 0$.

10.4 Dynamic Programming

The previous sections assembled the various pieces necessary to solve tasks such as the maze shown in figure 10.1. To recall, the essential difficulty is choosing an action at state **l**, since a reward or punishment is only received after a further action choice. This presents the problem of the delay in getting the reward or punishment from the experimenter which evaluates the action, and also, if the subsequent action choice is incorrect (going left rather than right at **s**), the problem that this evaluation maybe unhelpful.

A set of techniques from engineering called dynamic programming offer computationally tractable methods for solving these problems in fairly general cases called Markov decision problems. The method on which we concentrate is called policy iteration, which involves taking a policy such as that in equation 10.33, *evaluating* it, *ie* using temporal difference learning to learn the long term reward starting from state **u** using policy **p**, and finally using the evaluations to improve the policy. This process then repeats.

The appropriate context for prediction and action selection is that of stationary Markov decision problems, which idealise random worlds such as the maze in figure 10.1 in which the actions the animals takes have some bearing on the states (places) it visits and consequently the rewards it receives. A stationary Markov decision problem is defined by various quantities which are listed in table 10.5. The first is the state space $\mathcal{U}$, which is the set of all the available places in the world (we use u and u' as generic elements of $\mathcal{U}$). Second is the action space $\mathcal{A}$ (generic elements a and a'), which is set of all actions in the world. In general different states can permit different sets of actions, but we will ignore this. Third is a collection of transition matrices $P_{ua}[u']$, one for each action. The quantity $P_{ua}[u']$ is the probability that if the animal starts at state u and applies action a, then it will end up at state u'. Necessarily, $\sum_{u'} P_{ua}[u'] = 1$ for all u and a, since the animal has to end up somewhere. There can be absorbing states, which are u for which $P_{ua}[u] = 1$ for all actions a – *ie* there is no escape for the animal from these locations. The places marked '*' and '–' in figure 10.1 are absorbing. Finally, there are rewards r which might be stochastic, and could depend both on the state u and the action executed a. As before, we write $\langle r \rangle_u^a$ for the mean reward in this case. For convenience, we only consider Markov chains which are finite (*ie* $\mathcal{A}$ and $\mathcal{U}$ are both finite), absorbing, ie for which for all possible choices of actions, the animal will ultimately end up in one of the absorbing states, and in which the rewards are bounded. We also require that $\langle r \rangle_u^a = 0$ for all actions a at all absorbing states.

The elaboration of the representation $\mathbf{u}(t)$ into $\mathbf{u}^\tau(t)$ in equation 10.19 is required to make the prediction task appropriately Markovian. A particular characteristic of a Markov decision problem is that the distribution over the future state u' is independent of the states at past steps given the state u at the current step. This is only true for the prediction task for the mon-

Quantity	Definition
$\mathcal{U}$	state space
$\mathcal{A}$	action space
$P_{ua}[u']$	transition probability of going from u to u' under action a
$\langle r \rangle_u^a$	average reward for choosing action a at state u
$p_u[a]$	stochastic policy – the probability of choosing action a at state u

Table 10.5: Elements of a Markov decision problem.

key in figure 10.2 if it has some way of counting time between the stimulus and the reward. Otherwise, it would not be able to distinguish between two cases in which there are no stimuli present – one for which it has just received the stimulus, and so is expecting the reward, and the other for which it has just received the reward, and, if anything, is expecting the stimulus again.

The task for a system or animal facing a Markov decision problem starting in state u at time 0, is to choose actions $a(0), a(1)\ldots$ to maximise the expected sum total reward

$$v^*(u) = \max_{a(0),a(1),\ldots} \left\langle \sum_{t=0}^{\infty} r(t) \right\rangle \tag{10.49}$$

where $u(0)=u$. This sum depends not only on $\langle r \rangle_u^a$ directly, but also on the stochastic state sequence $u(t)$ that the actions arrange for the system. The state sequence is stochastic since $P_{ua}[u']$ need not be deterministic. $v^*(u)$ is called the optimal value function, since it relates for the best possible choice of actions the value of state u in terms of the amount of reward that will accrue starting from state u. Since the decision problem is Markov and stationary, the optimal value of state u does not depend explicitly on time.

The expectation over the infinite sum is rather awkward to handle, and the key insight in dynamic programming is the Bellman equation (Bellman, 1957):

$$v^*(u) = \max_{a(0),a(1),\ldots} \left\{ \left\langle r + \sum_{t=1}^{\infty} r(t) \right\rangle \right\} \tag{10.50}$$

$$= \max_{a(0)} \left\{ \langle r \rangle_u^{a(0)} + \max_{a(1),a(2),\ldots} \left\langle \sum_{t=1}^{\infty} r(t) \right\rangle \right\} \tag{10.51}$$

$$= \max_{a(0)} \left\{ \langle r \rangle_u^{a(0)} + \sum_{u'} P_{ua(0)}[u'] v^*(u') \right\} \tag{10.52}$$

where equation 10.52 follows from equation 10.51 by writing out the expectation over the transitions to all possible states u' at the first step starting at u, using the fact that the system is Markovian, and recognising that the maximisation is now over all the actions starting at u'. Intuitively, it says that maximising reward at u requires choosing the action $a(0)$ that maximises the sum of the mean immediate reward $\langle r\rangle_u^{a(0)}$ and the average of the largest possible values of all the states in which $a(0)$ would leave the system, weighted by their probabilities.

If $v^*(u)$ were known, for all states u, optimal actions would be easy to determine as a^* such that

$$v^*(u) = \langle r\rangle_u^{a^*} + \sum_{u'} P_{ua^*}[u']v^*(u') \ , \tag{10.53}$$

ie $v^*(u)$ creates a simple action choice problem, which only involves the single step (to u') from a difficult one, which involves multiple steps. Learning $v^*(u)$ is therefore an appropriate goal.

In policy iteration, the system maintains policy $\mathbf{p}$ which relates the probability $p_u[a]$ of choosing action a at state u. This policy could, of course, be deterministic. Policy iteration consists of two steps – policy evaluation, and policy improvement. Policy evaluation requires calculating for every state u the expected sum reward the system will receive, starting from state u and following policy $\mathbf{p}$. This quantity $v^{\mathbf{p}}(u)$ is defined as

$$v^{\mathbf{p}}(u) = \left\langle \sum_{t=0}^{\infty} r(t) \right\rangle_{\mathbf{p}} \tag{10.54}$$

where $u(0)=u$, and where $\langle\rangle_{\mathbf{p}}$ indicates that the expectation is now over the randomness in the policy as well as the transitions in the world. This can be viewed as an instance of the sum in equation 10.11, and temporal difference methods will be used to learn its value. By the same reasoning that led to the Bellman equation 10.52, this can be written as a set of self-consistency equations:

$$v^{\mathbf{p}}(u) = \sum_a p_u[a] \left\{ \langle r\rangle_u^a + \sum_{u'} P_{ua}[u']v^{\mathbf{p}}(u') \right\} \tag{10.55}$$

This is a finite set of purely linear equations, and so one could solve for $v^{\mathbf{p}}(u)$ using a matrix inverse operation. Reinforcement learning can be interpreted as a stochastic Monte-Carlo method for performing this matrix inversion (Barto & Duff, 1994).

Given the values $v^{\mathbf{p}}(u)$, policy iteration improves policy $\mathbf{p}$ by finding, for each state u, an action a^* which maximises the expression in the curly brackets in equation 10.55:

$$\langle r\rangle_u^a + \sum_{u'} P_{ua}[u']v^{\mathbf{p}}(u') \tag{10.56}$$

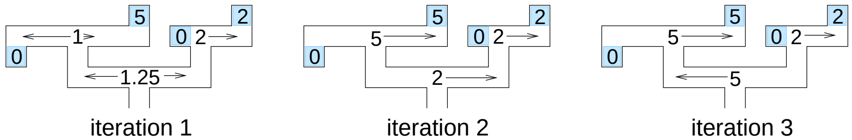

Figure 10.11: Policy iteration. The shaded boxes show the rewards at the terminal states in the maze, the relative lengths of the arrows show the probabilities of moving in the given directions, and the numbers at the choice points show the value of the associated policy. Even though the better action is initially favored at **l**, in iteration 2, the poor choice of action at **s** erases it, and it is not until iteration 3 that the optimal policy is found.

and making the new $p_u[a^*]=1$. One can show that the new policy will be better than the old policy, or equally good, if the old policy was already optimal. Further, since there is only a finite number of policies (since $\mathcal{U}$ and $\mathcal{A}$ are finite), policy iteration is bound to converge. Figure 10.11 shows an example of policy iteration in the maze. In this case, since the initial policy at state **s** is incorrect, going left and going right at state **l** appear to be equally bad. It is only once the value of state **s** has been appropriately found that the correct action can be chosen at **l**.

Even if the initial policy is stochastic, the new policy determined by equation 10.56 will be deterministic. In general, one can prove that there is a deterministic policy at least as good as any stochastic policy, and so one formally only needs to search through the space of deterministic policies. However, animals typically maintain stochastic policies – which achieve a smoother balance between exploration and exploitation.

In some cases, rewards and punishments received soon are more important than rewards and punishments received later. One natural way to accommodate this is a technique called *discounting*. For this, the goal for the policy is to maximise a discounted sum of future rewards, replacing equation 10.49 with

$$v^*(u) = \max_{a(0),a(1),\ldots} \left\langle \sum_{t=0}^{\infty} \gamma^t r(t) \right\rangle \tag{10.57}$$

where $0 \le \gamma \le 1$ is the discount factor. The smaller γ, the stronger the effect, *ie* the less important are temporally distant rewards. Exponential discounting is convenient, since it preserves the recursive nature of formulæ such as equation 10.52:

$$v^*(u) = \max_{a(0)} \left\{ \langle r \rangle_u^{a(0)} + \gamma \sum_{u'} P_{ua(0)}[u'] v^*(u') \right\} , \tag{10.58}$$

although behavioral evidence from instrumental conditioning studies suggests that hyperbolic discounting (replacing the expectation in equation 10.57 with something like $\langle \sum_{t=0}^{\infty} r(t)/(t+\theta) \rangle$ for a constant $\theta > 0$) might be more appropriate. Policy evaluation and improvement continue to work just as above, except that $P_{ua}[u']$ should be replaced by $\gamma P_{ua}[u']$.

Temporal-Difference Methods

Of course, policy evaluation and policy improvement all require complete knowledge of the quantities in table 10.5 that define the Markov chain, namely the transition matrices and the rewards. Indeed, one family of models is based on learning $P_{ua}[u']$ and $\langle r \rangle_u^a$ and using these values in more standard forms of dynamic programming. However, the insight in methods of reinforcement learning is that one can often use Monte-Carlo sampling to evaluate approximately sums such as $\sum_{u'} P_{ua}[u'] v^{\mathbf{p}}(u')$ in equation 10.56 that come from expectation operations. For instance if u' is picked from the distribution $P_{ua}[u']$, then

$$v^{\mathbf{p}}(u') \tag{10.59}$$

is an unbiased, albeit noisy, estimate of $\sum_{u'} P_{ua}[u'] v^{\mathbf{p}}(u')$. It is called a Monte-Carlo sample because of the randomness. It would be conventional to average together many such samples, all of which are independent. This is often tricky within a single trial – for instance, the animal may not be able to retrace its steps and retry many times all the actions available at a state – but can be performed across multiple trials.

Policy Evaluation

The temporal difference method for classical conditioning that we described above is a Monte-Carlo method for policy evaluation (Barto, Sutton & Watkins, 1989). Given a fixed (possibly stochastic) policy, the objectives expressed by equations 10.11 and 10.54 are the same. Suppressing, for the moment, the dependence on time t during a trial, consider the case for which the system starts at u, chooses action a according to $p_u[a]$, receives reward r, and moves to state u' chosen randomly according to $P_{ua}[u']$. Given an estimator $v(u)$ of $v^{\mathbf{p}}(u)$,

$$r + v(u') \tag{10.60}$$

is an exact Monte Carlo sample of

$$\sum_a p_u[a] \left\{ \langle r \rangle_u^a + \sum_{u'} P_{ua}[u'] v(u') \right\} \tag{10.61}$$

and consequently an approximate Monte Carlo sample of the expression on the right hand side of equation 10.55 (replacing $v^{\mathbf{p}}(u')$ with $v(u')$). Substituting the new notation, the temporal difference prediction error term from equation 10.15

$$\delta = r + v(u') - v(u) \tag{10.62}$$

can therefore be seen as an approximate Monte Carlo estimate of the difference $v^{\mathbf{p}}(u) - v(u)$, which is an appropriate error measure for training

$v(u)$ to equal $v^{\mathbf{p}}(u)$. Note that if the task uses discounted rewards (as in equation 10.57), then δ should be replaced by

$$r + \gamma v(u') - v(u) \ . \tag{10.63}$$

This change is appropriate for all the uses to which we turn δ.

Consider the case that we put back the dependence on time t within a trial (so $u = u(t)$, $a = a(t)$, $r = r(t)$, and $u' = u(t+1)$), and write $v(t) = v(u(t))$. Then, the expression in equation 10.15:

$$\delta(t) = r(t) + v(t+1) - v(t) \tag{10.64}$$

is the equivalent of δ in equation 10.62, and so is an appropriate measure of the error $v^{\mathbf{p}}(u(t)) - v(u(t))$. Finally, consider the case that we represent state $u(t)$ by a vector $\mathbf{u}(t)$ and the value $v(t) \equiv v(u(t)) = \mathbf{w} \cdot \mathbf{u}(t)$ (as in equation 10.10). Then, repeating equation 10.16, the standard delta rule for altering the weights in the light of this error is

$$\mathbf{w} \to \mathbf{w} + \epsilon \delta(t) \mathbf{u}(t) \ . \tag{10.65}$$

The link to Monte-Carlo estimates can be used to show that the estimates will converge such that

$$v(u) = v^{\mathbf{p}}(u) \qquad \text{for all states } u, \tag{10.66}$$

provided that suitable conditions on the learning rates ϵ and on the whole set of state representations $\mathbf{u}$ are met. Thus, temporal difference learning accomplishes policy evaluation.

The temporal difference model therefore has its underpinnings as a correct way of doing Monte Carlo policy evaluation, using samples of transitions and rewards from the world rather than requiring direct knowledge of $\langle r \rangle_u^a$ or $P_{ua}[u']$. We introduced trace terms (equation 10.17)

$$\tilde{\mathbf{u}}(t) = \tilde{\mathbf{u}}(t-1) + (1-\lambda)(\mathbf{u}(t) - \tilde{\mathbf{u}}(t-1)) \tag{10.67}$$

in order to arrange for learning in special cases in which the prediction problem is not Markovian, for instance, if the reward arrives a set time after the light but without there being any external change of state. However, even if the problem is Markovian, for instance, if the elaborated representation of the light in equation 10.19 is used, then trace terms in the context of temporal difference models

$$\mathbf{w} \to \mathbf{w} + \epsilon \delta(t) \tilde{\mathbf{u}}(t) \ . \tag{10.68}$$

can also be shown to be admitted into the canon of provably correct prediction rules. This particular learning rule is called the TD(λ) rule (Sutton, 1988) where λ is the time constant for the trace rule. Stimulus traces hasten learning in many cases.

If the representation of $v(u)$ were more complicated, *eg* some more general non-linear form $v(u; \mathbf{w})$, then learning rule 10.65 would become

$$\mathbf{w} \to \mathbf{w} + \epsilon \delta(t) \nabla_{\mathbf{w}} v(u(t); \mathbf{w}) \ , \tag{10.69}$$

although the guarantees of appropriate convergence would be lost.

Policy Improvement

The reinforcement learning version of policy improvement step is based on similar principles. As discussed above, the requirement for improving a policy is finding the action a^* that maximises the expression in equation 10.56. In fact, any policy $\mathbf{p}'$ that improves the average value of

$$\sum_a p'_u[a] \left\{ \langle r \rangle_u^a + \sum_{u'} P_{ua}[u'] v^{\mathbf{p}}(u') \right\} . \tag{10.70}$$

for every state u will be guaranteed to be a better policy. The point of using $\mathbf{p}'$ rather than $\mathbf{p}$ is that policy improvement requires finding actions that improve the average reward whilst *assuming* that the terms $v^{\mathbf{p}}(u')$ are fixed. The idea for a single state u is to treat equation 10.70 as an example of equation 10.34, replacing $\langle r \rangle^a$ there by $\langle r \rangle_u^a + \sum_{u'} P_{ua}[u'] v^{\mathbf{p}}(u')$ to take long term as well as immediate reward into account.

In equation 10.35, r is used as a Monte-Carlo sample of $\langle r \rangle^a$ in order to perform stochastic gradient ascent in the average reward in equation 10.34. Similarly, starting from state u, applying action a, receiving reward r and ending up in state u', the RL policy improvement algorithm uses

$$r + v(u') \tag{10.71}$$

as an approximate Monte-Carlo sample to perform stochastic gradient ascent in the average reward in equation 10.70. Again, the approximation lies in using $v(u')$ instead of $v^{\mathbf{p}}(u')$. The value of $\bar{r}$ in equation 10.35 is arbitrary, but is usually taken as the average reward. In this case, $v^{\mathbf{p}}(u)$ is exactly the average value in equation 10.70 for state u, $v(u)$ is the estimate of $v^{\mathbf{p}}(u)$, and so the temporal difference error from equation 10.15

$$\delta = r + v(u') - v(u) \tag{10.72}$$

is the approximate equivalent of the term $r - \bar{r}$ in equation 10.35, and so forms the basis of an appropriate policy improvement method. The advantage of using this form is that a single prediction error signal δ trains both the predictions and the actions. Crudely, $\delta(t)$ will be positive if *either* the prediction $v(u(t))$ is too low, *or* the action a tried is much better than average (or possibly both). The portion of the system that estimates $v(u)$ is called the *critic*, since its output is used to criticise actions

If states are represented by vectors $\mathbf{u}$, then the softmax representation of the policy $\mathbf{p}$ in equation 10.41, using action vectors $\mathbf{S}$ is standard. In this case, the policy improvement rule becomes

$$\mathbf{s}^b \rightarrow \mathbf{s}^b + \eta \delta(t)(\delta_{ab} - p_{\mathbf{u}(t)}[b])\mathbf{u}(t) , \tag{10.73}$$

again putting back the dependence on time t within a trial. By contrast with the critic, the portion of the network that represents the policy is called the *actor*. The overall model is sometimes called the actor-critic (Barto, Sutton & Anderson, 1983).

There are two potential problems with this implementation of policy iteration. The first is that equation 10.73 offers the possibility of trading off improvement in the policy at one state for a worsening at other states. This can be prevented by using representations of states $\mathbf{u}$ that are orthogonal, but this is a rather draconian requirement. The second is that there is an interaction between the stochasticity in the reinforcement learning versions of policy evaluation and policy improvement. This means that it is not yet clear theoretically if the two together are guaranteed to converge. One could perform temporal difference policy evaluation (which does provably converge) until convergence before attempting policy improvement, and this would be sure to work. The Q-learning method we discuss below, which is an instance of value iteration, avoids this problem.

In the experiments on primate conditioning, a population of cells in the substantia nigra, another area containing dopaminergic cells, were also found to have similar behavior to those shown in figure 10.2. Whereas the cells in the ventral tegmental area project to the nucleus accumbens, those in the substantia nigra project to the dorsal striatum. This is a large network that is believed to be involved in the selection and sequencing of actions in virtue of its connections with somatosensory and motor processing areas of cortex, and also its modes of failure. For instance, Parkinson's disease is caused by degeneration of the dopamine cells in the substantia nigra that project to the striatum, and exhibits characteristic symptoms indicating problems with motor control such as the inability to initiate voluntary movement and rigidity that arises from the simultaneous activation of extensor and flexor muscles. Under the theory, this projection also reports $\delta(t)$ in order to control the choice of actions, and also the learning of this choice.

Although reinforcement learning methods are theoretically attractive, their use as models of animal learning faces two major concerns. One is that Markov decision problems might seem rather abstract – it is not clear that they are reasonable models of tasks that animals face, or that they accurately reflect the substantial prior knowledge that animals bring to bear on solving more ethologically apposite problems. However, foraging for food and water actually poses many problems that can be seen in terms of Markov decision problems and their extensions (*eg* Mangel & Clark, 1988), so reinforcement learning provides a computational link between ethological and psychological experiments. The other concern is that reinforcement learning is generally rather slow. However, this is quite dependent on the stimulus representations used – good representations lead to fast learning.

As an example of the reinforcement learning solution of a less abstract task, figure 10.12 shows its application to the water maze. This is a navigation problem in which rats learn the location of a platform hidden under the surface of a circular pool of milky water. Figure 10.12A shows the

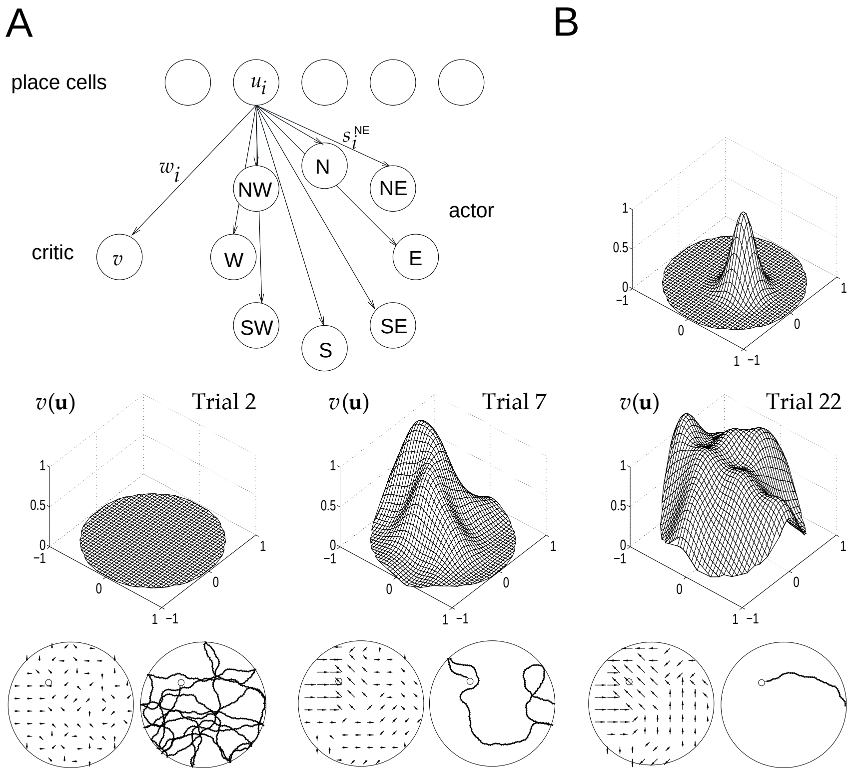

Figure 10.12: Reinforcement learning model of a rat solving a simple water maze task in a $2m$ diameter circular maze. A) There are 493 place cell inputs and 8 actions. The rat moves at $0.3ms^{-1}$ and reflects off the walls of the maze if it hits. B) Gaussian place field for a single input cell with width $\sigma = 0.16m$; the centers of the place fields are uniformly distributed across the pool. C) Upper: the development of the value function $v(\mathbf{u})$ over the first 20 trials, starting from $v(\mathbf{u}) = 0$. Lower arrow plots: the action a^* with the largest value of $\mathbf{s}^{a^*} \cdot \mathbf{u}$ for each location in the maze; the small circles show the goal. Lower path plots: the actual path taken by the rat from a random starting point to the goal. Adapted from Foster, Morris & Dayan (1998).

structure of the model, with the state representation $\mathbf{u}$ providing input to the critic and a collection of 8 possible actions for the actor. Figure 10.12B shows the activation of one of the input units as a function of spatial position in the pool. This is a crude model of the activity of a hippocampal place cell, a cell in area CA1 of the hippocampus whose activity is spatially restricted. Figure 10.12C shows the course of learning (trials 1, 5 and 20) of the value function (upper figures) and the policy (lower figures with arrows; the goal is marked by the small circle), and also sample paths. A reward of 1 is given at the goal, and is discounted (according to equation 10.57), modeling the incentive for the rat to find the goal as soon as possible. Each trial consists of starting the rat from a random location at the outside of the maze and letting it run to find the goal. The final value function (at trial 20) is rather inaccurate, but, nevertheless, the policy learned is broadly correct, and the paths become quite short. As judged by measures such as path length, learning proceeds in the model at a rate comparable to that of actual rats. Learning is fast because the representation of the maze afforded by place cells is ideal – making similar nearby places, which are similar distances from the goal and require similar actions. Rats certainly learn more about the structure of the maze than does the model – nevertheless the model shows something of the power of a primitive, but general, learning method.

Value Iteration and Q-Learning

Value iteration is a dynamic programming method that is an alternative to policy iteration. For one version of this, a collection of values $Q(u,a)$ is maintained, one for each action a at each state u, and is updated according to

$$Q(u,a) \rightarrow \langle r \rangle_u^a + \sum_{u'} P_{ua}[u'] \max_{a'} Q(u',a') \tag{10.74}$$

If this is performed for all actions and all states, then it is possible to show that $Q(u,a)$ will converge to the optimal functions $Q^*(u,a)$ which are defined by the self-consistent equations:

$$Q^*(u,a) = \langle r \rangle_u^a + \sum_{u'} P_{ua}[u'] \max_{a'} Q^*(u',a') \tag{10.75}$$

and are linked to the optimal value function in equation 10.49 by

$$v^*(u) = \max_{a'} Q^*(u,a') \tag{10.76}$$

Given the optimal $Q^*(u,a)$ values, an optimal (deterministic) policy is defined by choosing any action a^* at state u such that

$$Q^*(u,a^*) = \max_{a} Q^*(u,a) \; . \tag{10.77}$$

The values $Q(u,a)$ before convergence can always be used to define a policy (as in equation 10.77), but there is no reason to believe that it will be an *optimal* policy.

As before, implementing the value iteration operation in equation 10.74 requires that $\langle r \rangle_u^a$ and $P_{ua}[u']$ be known. The reinforcement learning method called $Q-$learning (Watkins, 1989) avoids this by using sampling. In this case, based on starting from state u, taking action a, receiving reward r, and ending up at state u;,

$$r + \max_{a'} Q(u',a') \tag{10.78}$$

is used as a Monte-Carlo sample of the right hand side of equation 10.74. The standard delta learning rule for $Q(u,a)$ in this case is

$$Q(u,a) \to Q(u,a) + \epsilon(r + \max_{a'} Q(u',a') - Q(u,a)) \qquad (10.79)$$

where ϵ is a learning rate, which implements equation 10.74 by averaging over multiple random examples.

$Q-$learning only involves the estimation of one quantity rather than the two of policy iteration. It is therefore possible to prove that under a set of conditions, including the fact that every action is tried at every state an infinite number of times, and with a decreasing learning rate ϵ, that using equation 10.79, $Q(u,a)$ is almost certain to tend to the optimal $Q^*(u,a)$, thus finding the optimal policy.

10.5 Further Reading and References

Further Reading

Bertsekas, DP & Tsitsiklis, JN (1996) *Neuro-Dynamic Programming*. Belmont, MA:Athena Scientific.

Bush, RR & Mosteller, F (1955) *Stochastic Models for Learning*. New York:Wiley.

Dickinson, A (1980) *Contemporary Animal Learning Theory*. Cambridge: Cambridge University Press.

Gabriel, M & Moore, JW, editors (1989) *Learning and Computational Neuroscience*. Cambridge, MA:MIT Press.

Gallistel, CR (1990) *The Organization of Learning*. Cambridge, MA:MIT Press.

Grossberg, S, editor (1988) *Neural Networks and Natural Intelligence*. Cambridge, MA:MIT Press.

Mackintosh, NJ (1983) *Conditioning and Associative Learning*. Oxford:Oxford University Press.

Schultz, W, Dayan, P & Montague, PR (1997) A neural substrate of prediction and reward. *Science* **275**:1593-1599.

Shanks, DR (1995) *The Psychology of Associative Learning.* Cambridge: CUP.

Sutton, RS & Barto, AG (1998) *Reinforcement Learning.* Cambridge, MA: MIT Press.

References

Barto, AG, Sutton, RS & Anderson, CW (1983) Neuronlike elements that can solve difficult learning problems. *IEEE Transactions on Systems, Man, and Cybernetics* **13**:834-846.

Barto, AG, Sutton, RS & Watkins, CJCH (1990) Learning and sequential decision making. In M Gabriel & J Moore, editors, *Learning and Computational Neuroscience: Foundations of Adaptive Networks*. Cambridge, MA:MIT Press, Bradford Books.

Berry, DA & Fristedt, B (1985) *Bandit Problems: Sequential Allocation of Experiments.* New York, NY: Chapman and Hall.

Blackburn, JR, Pfaus, JG & Phillips, AG (1992) Dopamine functions in appetitive and defensive behaviours. *Progress in Neurobiology* **39**:247-279.

Dickinson, A (1994) Instrumental conditioning. In NJ Mackintosh, editor, *Animal learning and cognition. Handbook of perception and cognition series 3* San Diego, CA: Academic Press:45-79.

Friston, KJ, Tononi, G, Reeke, GN Jr, Sporns, O & Edelman, GM (1994) Value-dependent selection in the brain: simulation in a synthetic neural model. *Neuroscience* **59**229-243.

Gallistel, CR & Gibbon, J (1998) *Time, Rate and Conditioning.* Forthcoming.

Gibbon, J, Baldock, MD, Locurto, C, Gold, L & Terrace, HS (1977) Trial and intertrial durations in autoshaping. *Journal of Experimental Psychology: Animal Behavior Processes* **3**:264-284.

Gibbon, J, Farrell, L, Locurto, CM, Dncan, HJ & Terrace, HS (1980) Partial reinforcement in autoshaping with pigeons. *Animal Learning and Behavior* **8**:45-59.

Gluck, MA, Reifsnider, ES & Thompson, RF (1990) Adaptive signal processing and the cerebellum: Models of classical conditioning and VOR adaptation. In MA Gluck & DE Rumelhart, editors, *Neuroscience and Connectionist Theory. Developments in Connectionist Theory*. Hillsdale, NJ:Erlbaum, 131-185.

Gluck, MA & Thompson, RF (1987) Modeling the neural substrates of associative learning and memory: a computational approach. *Psychological Review* **94**:176-191.

Grossberg, S & Schmajuk, NA (1989) Neural dynamics of adaptive timing and temporal discrimination during associative learning. *Neural Networks* **2**:79-102.

Houk, JC, Davies, JL & Beiser, DG, editors (1995) *Models of Information Processing in the Basal Ganglia*. Cambridge, MA:MIT Press.

Konorski, J (1948) *Conditioned Reflexes and Neuron Organization*. Cambridge: Cambridge University Press.

Mackintosh, NJ (1975) A theory of attention: Variations of the associability of stimuli with reinforcement. *Psychological Review* **82**:276-298.

Mangel, M & Clark, CW (1988) *Dynamic Modeling in Behavioral Ecology*. Princeton, NJ:Princeton University Press.

Montague, PR, Dayan, P & Sejnowski, TJ (1996) A framework for mesencephalic dopamine systems based on predictive Hebbian learning. *Journal of Neuroscience* **16**:1936-1947.

Narendra, KS & Thatachar, MAL (1989) *Learning Automata: An Introduction*. Englewood Cliffs, NJ:Prentice-Hall.

Pearce, JM & Hall, G (1980) A model for Pavlovian learning: Variations in the effectiveness of conditioned but not of unconditioned stimuli. *Psychological Review* **87**:532-552.

Rescorla, RA & Wagner, AR (1972) A theory of Pavlovian conditioning: The effectiveness of reinforcement and non-reinforcement. In AH Black & WF Prokasy, editors, *Classical Conditioning II: Current Research and Theory*. New York:Aleton-Century-Crofts, 64-69.

Schultz, W, Apicella, P & Ljungberg, T (1993) Responses of monkey dopamine neurons to reward and conditioned stimuli during successive steps of learning a delayed response task. *Journal of Neuroscience* **13**:900-913.

Schultz, W (1992) Activity of dopamine neurons in the behaving primate. *Seminars in the Neurosciences* **4**:129-138.

Solomon, RL & Corbit, JD (1974) An opponent-process theory of motivation. I. Temporal dynamics of affect. *Psychological Review* **81**:119-145.

Sutton, RS (1988) Learning to predict by the methods of temporal difference. *Machine Learning* **3**:9-44.

Sutton, RS & Barto, AG (1989) Time-derivative models of Pavlovian conditioning. In M Gabriel & JW Moore, editors, *Learning and Computational Neuroscience*. Cambridge, MA:MIT Press.

Watkins, CJCH (1989) *Learning from Delayed Rewards*. PhD Thesis, University of Cambridge, Cambridge, UK.

Williams, RJ (1992) Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning* **8**:229-256.

Wise, RA (1982) Neuroleptics and operant behavior: The anhedonia hypothesis. *Behavioral and Brain Sciences* **5**:39-87.

10.6 Exercises

1. Implement acquisition and extinction as in figure 10.3 using the Rescorla-Wagner (delta) rule (equation 10.4).

2. Add an extra conditioned stimulus and demonstrate the delta rule's proficiency at blocking and failure to exhibit secondary conditioning.

3. Consider the case of partial reinforcement (studied in figure 10.8 in which the reward is only provided on some (systematic or random) fraction of times the conditioned stimuli are delivered. Show acquisition and extinction curves for the the case of simple Pavlovian conditioning using the Rescorla-Wagner rule as a function of trials and numbers of reinforcements for different reinforcement schedules. Do they match the empirical data?

4. What is a natural statistical interpretation of the learning rate ϵ for the Rescorla-Wagner rule? Under this interpretation, what should happen to ϵ, and what effect would this have on the acquisition and extinction curves?

5. Implement the temporal difference learning model with the trace rule, as in equation 10.18, and show that learning traces as in figure 10.5B result. Demonstrate that and justify why the trace model nevertheless shows secondary conditioning? Show what happens to the weights and to $\delta(t)$ if there are two perfectly reliable predictors of reward, one of which always precedes the other and the consequences of making either or both unreliable in their occurrence on each trial and/or the time before the reward they arise. Explain the results.

6. Implement the full temporal difference learning model of figure 10.7. Show what happens in the case that there are two reliable or unreliable predictors as in exercise 5.

7. Implement a stochastic three-armed bandit in which the choice of arm a leads to a reward with a probability p_a. Implement the stochastic learning automaton of equations 10.33 and 10.36 and demonstrate its capacity to find the best arm. If the probabilities of reward p_a are allowed to change, compare the effect of different values of η by averaging over long learning runs.

8. Implement the maze of figure 10.1 with a reward of 1 at the location marked '*', rewards of -1 at the location marked '–', and no rewards at other locations. Using a representation in which each state $\mathbf{u}^a$, $\mathbf{u}^b$ and $\mathbf{u}^c$ has an orthogonal representation (*eg* $\mathbf{u}^a = (1, 0, 0)$, $\mathbf{u}^b = (0, 1, 0)$, $\mathbf{u}^c = (0, 0, 1)$), implement the full actor-critic method of dynamic programming (equations 10.65 and 10.73) and show how it finds the optimal policy.

9. What is the effect of using a state representation in which the representations of the state are not orthogonal – can it make temporal difference learning faster? Why?