Reinforcement learning and conditioning: an overview

Jérémie Jozefowiez
Université Charles de Gaulle, Lille, France
jozefowiez@univ-lille3.fr

October 8, 2002

Contents

Chapter 1

Computational foundations of reinforcement learning

Contingencies are dependency relations among elements in the environment of an organism. For instance, in our everyday environment, a traffic light usually turns green a few time after it has turned red or, in a baby's environment, crying often brings the appearance of the parents. If all the elements involved in a contingency are stimuli (as in the traffic light example), it is a *Pavlovian contingency*. Otherwise (as in the baby example), it is an *operant contingency*.

The environment of an organism could be described as a set of Pavlovian and operant contingencies. When new contingencies are added or old ones are modified, we usually observe a modification in the behavior of the organism: this is the phenomena that we call adaptation. Behavior is fundamentally adaptive in the sense that it is sensible to modifications in the environmental contingencies. Since an animal could not survive in its environment without this ability to tune its activity to its environment, adaptation is certainly the most important feature of behavior and the key to its understanding: no matter the mechanisms underlying behavior, they have been selected during evolution for their capacity to generate adaptive behavior.

In experimental psychology, the two main procedures for the study of adaptive behavior are *Pavlovian and operant conditioning* (Pavlov, 1927; Skinner, 1938). Pavlovian conditioning studies the way organism modifies their activity when exposed to Pavlovian contingencies while operant conditioning does the same for operant contingencies. These procedures has allowed to collect an impressive amount of data on the way animals adapt to their environment. But what are the processes involved? In this text, we will point out the relevance of so-called *reinforcement learning* algorithms which have been developed in computer science. These methods for learning

in artificial autonomous agents are prediction methods solving a special kind of optimization problems called *Markovian decision problems* (MDP). The next section provide an introduction to MDP and reinforcement learning and discuss the relation between them and artificial neural network. We will then review their applications in the conditioning literature.

1.1 Markovian decision problems

A MDP is composed of an environment, characterized by a set of states $s \in S$ and of an agent characterized by a set of controls (or actions) $u \in U$. The agent is further characterized by a policy π which attributes to each $s \in S$ a control $\pi(s) \in U$. This kind of policy is deterministic. Policies can also be stochastic but, most of the time, we will consider only deterministic ones.

The agent and the environment interact together according to the following sequence

1. At time t, the environment is in state $s(t) = s$.
2. The agent emits control $u(t) = \pi[s(t)] = u$.
3. The environment goes into state $s(t+1) = s'$.
4. The agent collects an immediate amount of primary value $r(t) = r$.

The dynamics of the environment is *Markovian*: $s(t+1)$ only depend upon $s(t)$ and $u(t)$, the states previously visited and the controls previously emitted have no impact on the current dynamics of the environment. So, we can define the *state transition probability* $p(s'|s,u)$ which is the probability that $s(t+1) = s'$ when $s(t) = s$ and $u(t) = u$. The Markovian property also holds for $r(t)$ whose *mean* value is determined by the *return function* $f[s(t), s(t+1), u(t)]$.

The Markov property allows the computation of $V^\pi(s)$, the *situation value of state s for policy* π. It is the total (discounted) amount of primary value that an agent can expect to collect if, while the environment is in state s, it begins to follow policy π.

$$\begin{aligned} V^\pi(s) &= \sum_u p(u|s) \sum_{s'} p(s'|s,u)[f(s,s',u) + \gamma V^\pi(s')] \\ &= \sum_u p(u|s) Q^\pi(s,u) \end{aligned} \tag{1.1}$$

This is *Bellman's equation for policy* π. $p(u|s)$ is the probability of emitting control u when the environment is in state s. It is, of course, determined by

π. If π is deterministic, then $p(u|s) = 1$ if $u = \pi(s)$ and 0 otherwise. γ is a free positive parameter always smaller than 1 called the *discount factor* which controls the impact of delayed rewards on the value $V^\pi(s)$. The closer it is to 1, the more they are taken into account. Finally, $Q^\pi(s, u)$ is the *state-action value of state s and action u for policy π* (or, more shortly, the *Q-value*). It is the total (discounted) amount of reward that an agent can expect to collect if, while the environment is in state s, it emits control u and then follows policy π. Note that, for a deterministic policy, $V^\pi(s) = Q^\pi[s, \pi(s)]$.

Equation (1.1) defines the function V^π called the *value function for policy* π. We have

$$\begin{aligned} V^\pi &: S \to \Re \\ &: s \mapsto V^\pi(s) \end{aligned} \tag{1.2}$$

A related function is Q^π, the *state-action value function for policy* π defined as

$$\begin{aligned} Q^\pi &: S \times U \to \Re \\ &: s, u \mapsto Q^\pi(s, u) \end{aligned} \tag{1.3}$$

In a MDP, the goal of the agent is to find an *optimal policy* π^* which maximizes the total (discounted) amount of primary value that the agent can collect in any state of the environment. Put it in another way, it is to find a policy π^* whose value function $V^{\pi^*}$is equal to V^*, the *optimal value function* defined as

$$\begin{aligned} V^* &: S \to \Re \\ &: s \mapsto V^*(s) = \max_\pi V^\pi(s) \end{aligned} \tag{1.4}$$

A related function is Q^*, the *optimal state-action value function* defined as

$$\begin{aligned} Q^* &: S \times U \to \Re \\ &: s, u \mapsto Q^*(s, u) = \max_\pi Q^\pi(s, u) \end{aligned} \tag{1.5}$$

Although there can be several optimal policies, the optimal value function V^* is unique. So, since selecting the action with the highest Q*-value is necessarily an optimal policy, we have

$$\begin{aligned} V^*(s) &= \max_u Q^*(s, u) \\ &= \max_u \max_\pi Q^\pi(s, u) \\ &= \max_u \max_\pi \sum_{s'} p(s'|s, u)[f(s, s', u) + \gamma V^\pi(s')] \end{aligned}$$

$$
\begin{aligned}
&= \max_u \sum_{s'} p(s'|s,u)[f(s,s',u) + \gamma \max_\pi V^\pi(s')] \\
&= \max_u \sum_{s'} p(s'|s,u)[f(s,s',u) + \gamma V^*(s')] \qquad (1.6)
\end{aligned}
$$

Equation (1.6) is *Bellman's optimality equation* (see Bertsekas, 1995; Bertsekas & Tsitsiklis, 1996 and Sutton & Barto, 1998 for further details).

1.2 Solving Markovian decision problems

Algorithms allowing an agent to solve a MDP use Bellman's optimality equation to compute the optimal value function. An optimal policy is then derived from it. They have been labeled *reinforcement learning (RL) algorithms* in artificial intelligence and are subdivided into two groups: *dynamic programming* on one side and *neurodynamic programming* on the other.

1.2.1 Dynamic programming

Dynamic programming (DP) requires an explicit knowledge of the environment e.g. of the state transition probabilities and of the return function. The optimal value function and the optimal policy are computed off-line, before any interaction between the agent and its environment.

What are the fundamentals of DP? Up to now, we have only considered MDP with an *infinite horizon* e.g. MDP where the agent and the environment interact for an infinite number of time steps. Now, consider a MDP with a horizon of 1: the agent and the environment only interact for one time step. If we note V_1^* the optimal value function for that problem, we have (using the same reasoning as the one for the derivation of equation 1.6)

$$
\begin{aligned}
V_1^*(s) &= \max_u Q_1^*(s,u) \\
&= \max_u \max_\pi Q_1^\pi(s,u) \qquad (1.7) \\
&= \max_u \sum_{s'} p(s'|s,u)[f(s,s',u) + \gamma V_0^*(s')]
\end{aligned}
$$

Where $V_0^*(s)$ is a fixed terminal reward (it can be set to 0). Now, consider the same MDP problem but with a horizon of 2. Using the same reasoning as above, we have

$$
\begin{aligned}
V_2^*(s) &= \max_u Q_2^*(s,u) \\
&= \max_u \max_\pi Q_2^\pi(s,u) \qquad (1.8) \\
&= \max_u \sum_{s'} p(s'|s,u)[f(s,s',u) + \gamma V_1^*(s')]
\end{aligned}
$$

Where $V_1^*(s)$ is given by equation (1.7). The same way, we have

$$\begin{aligned} V_3^*(s) &= \max_u Q_3^*(s,u) \\ &= \max_u \max_\pi Q_3^\pi(s,u) \\ &= \max_u \sum_{s'} p(s'|s,u)[f(s,s',u) + \gamma V_2^*(s')] \end{aligned} \tag{1.9}$$

And, more generally, if the optimal value function for the MDP with a horizon of n is known, we have

$$V_{n+1}^*(s) = \max_u \sum_{s'} p(s'|s,u)[r + \gamma V_n^*(s')] \tag{1.10}$$

Equation (1.10) is called *Bellman's principle of optimality.* It can be used to generate a sequence of optimal value functions $V_1^*, V_2^*, ..., V_i^*, ..., V_n^*$. Since it can be shown that

$$V^*(s) = \lim_{n\to\infty} V_n^*(s) \tag{1.11}$$

This sequence is assured to converge on V^*.

This algorithm, called *value iteration*, is one of the main DP methods. Other DP algorithms rely on similar principles as those described in equations (1.10) and (1.11) (see Bertsekas, 1995 for further details).

Once the optimal value function is known, it is easy to derive an optimal policy from it: for a given state $s \in S$, the agent has simply to select the control with the highest $Q^*(s,u)$ (see equation 1.6). So, most of the time, only deterministic policies are optimal. The only exception is when several controls have the same $Q^*(s,u)$. In such a case, choosing any one of these controls is optimal and so, a stochastic policy can be optimal. But, in all other cases, stochastic policies are suboptimal because they imply the selection of a control which has not the highest $Q^*(s,u)$.

1.2.2 Neurodynamic programming

Neurodynamic programming (NDP) requires no a priori knowledge about the environment dynamics and the optimal value function is computed on line, while the agent is interacting with its environment.

These algorithms store estimations v^π, v^*, q^πor q^* (of respectively V^π, V^*, Q^πor Q^*) for each state of the environment and, possibly, for each control of the agent. The interaction with the environment is used as a way to prompt the environment in order to have access to better evaluation ψ of either of the functions they try to estimate. So, if the agent is storing evaluations v of V^π or V^* or q of Q^π or Q^*, it will prompt the environment which will return

an estimation $\psi(s)$ of $V^\pi(s)$ or $V^*(s)$ or $\psi(s,u)$ of $Q^\pi(s,u)$ or $Q^*(s,u)$. The agent will use these estimations to improve its own evaluations using one of these two following equations:

$$\begin{aligned} v(s) &:= v(s) + \alpha[\psi(s) - v(s)] \\ q(s,u) &:= q(s,u) + \alpha[\psi(s,u) - q(s,u)] \end{aligned} \tag{1.12}$$

where α is a learning rate.

For instance, consider Q-learning (Watkins & Dayan, 1992), the most popular one of these algorithms and the NDP equivalent of value iteration. Q-learning directly tries to evaluate Q^* by storing estimations q^* which are updated on a real-time basis according to the following equation

$$q^*[s(t),u(t)] := q^*[s(t),u(t)] + \alpha\{r(t) + \gamma \max_u q^*[s(t+1),u] - q^*[s(t),u(t)]\} \tag{1.13}$$

So, in Q-learning, $\psi[s(t),u(t)] = r(t) + \gamma \max_u q^*[s(t+1),u]$ which is indeed the best immediate evaluation of $Q^*[s(t),u(t)]$ the agent can access at time $t+1$ (see equation 1.10).

1.2.3 Policy design in neurodynamic programming

It has been proved (Watkins & Dayan, 1992) that, if the agent and the environment interact for an infinite amount of time and if the learning rate is a decreasing function of time, then Q-learning converges to Q^* *no matter the policy actually followed by the agent* given that any state-action pair (s,u) is sufficiently visited.

It means that the agent must use a stochastic policy. But, as we said, stochastic policies are most of the time suboptimal. This is the *exploration/exploitation dilemma* faced by all the agents using NDP (Sutton & Barto, 1998). They must choose between exploiting, e.g. using an optimal deterministic policy, and exploring, e.g. using a non-optimal stochastic policy. In the first case, they will not improve their knowledge of the value function and so, will be unable to improve their performance. In the second one, they will act sub-optimally.

Solutions to this dilemma are not handled by NDP which are mere prediction methods but by a mechanism deriving the agent's policy from the computation realized by the NDP algorithm. The design of such a mechanism is currently more an art than a science (Sutton & Barto, 1998).

If the agent is storing the Q-values for each state of the environment and for each control (if it is using Q-learning, for instance), two popular solutions are *ε-greedy policies* and *Boltzmann exploration* (also called *softmax action selection).*

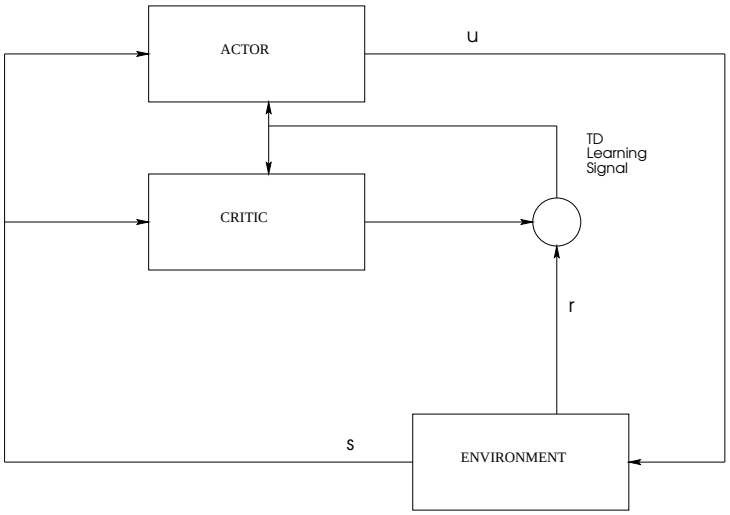

Figure 1.1: General organization of an actor/critic architecture

A greedy policy is simply a policy which picks the control with the highest Q*-value. An ε-greedy policy does exactly the same except that there is a small probability ε on each trial that it picks the control at random. In this way, the agent's behavior stays close to an optimal deterministic policy while keeping enough variability to improve its knowledge of the optimal value function.

A disadvantage of an ε-greedy policy is that it picks the controls at random, making no distinction between controls for which the prediction of reward is poor and controls for which it is higher. *Boltzmann exploration* does this. The following equation

$$p(u'|s) = \frac{e^{Q(s,u')/T}}{\sum_{u \in U} e^{Q(s,u)/T}} \tag{1.14}$$

is used to determine the probability of emission of a control in a given state. T is a free parameter called the *temperature* of the system. The higher it is, the more stochastic control selection is. On the other hand, when $T \to 0$, only the control with the highest Q-value is emitted. During the initial stage of training, T is set to a high value, hence favoring exploration. As learning progresses, it is slowly decreased, favoring more and more exploitation until it comes close to 0 and the agent learns no more (Sutton & Barto, 1998).

Another popular solution is a special kind of agent architecture called an *actor/critic architecture* (Barto, Sutton, & Anderson, 1983; Sutton & Barto, 1998). As it is shown in Figure 1.1, it is composed of two parts.

- First, there is the critic which uses $TD(\lambda)$, a NDP algorithm designed to evaluate V^π (Sutton, 1988; Sutton & Barto, 1998). Its learning equation is

$$v^\pi(s) := v^\pi(s) + \alpha \{r(t) + \gamma v^\pi[s(t+1)] - v^\pi[(s(t)]\} e_\lambda(s,t) \quad (1.15)$$

which is just the standard NDP equation applied to the evaluation of V^π with $\psi[s(t)] = r(t) + \gamma v^\pi[s(t+1)]$. This is very similar to the ψ term used in Q-learning. This is not surprising because Q-learning and $TD(\lambda)$ both belong to a wider family of NDP algorithms called *temporal difference methods* (Sutton & Barto, 1998) which all use about the same kind of ψ. Their main characteristic is that they update their prediction on a real-time basis which is not the case for all the NDP methods (see, for instance, the *Monte Carlo methods* in Sutton & Barto, 1998).
The only new thing here is $e_\lambda(s,t)$, the *eligibility trace* of state s at time t. The eligibility trace of a state increases when a state has been visited (e.g. when $s(t) = s$) and decreases otherwise (e.g. when $s(t) \neq s$). As long as it is positive, the value of the state can be updated as shown in the above equation. In this way, states recently visited can get some credit for the current amount of reward collected (see Sutton & Barto, 1998 for a fuller discussion of eligibility traces and especially of the way they fill the gap between temporal difference methods and Monte Carlo techniques). The decay of an eligibility trace is controlled by a single free parameter called λ (this is actually the λ term appearing in the name $TD(\lambda)$). The higher λ, the slower the decay of the eligibility trace. When $\lambda = 0$, only the state just visited is eligible for modification. There are several formulas describing how λ controls the temporal evolution of eligibility traces. For instance, Singh and Sutton (1996) have studied the property of so-called *replacing* eligibility traces whose temporal evolution is governed by the following equation

$$e_\lambda(s,t) = \begin{cases} 1 & if \quad s = s(t) \\ \gamma\lambda e_\lambda(s,t-1) & if \quad s \neq s(t) \end{cases} \quad (1.16)$$

Other formulas exist (see Sutton & Barto, 1990, 1998 for some examples). Which one is the most appropriate depends on the characteristics of the MDP the agent has to solve. Note that eligibility traces is a tool that can be added to any temporal difference method. So, for instance, adding eligibility traces to Q-learning, we get a new algorithm, $Q(\lambda)$, whose learning equation is

$$q^*(s,u) := q^*(s,u) + \alpha\{r(t) + \gamma \max_u q^*[s(t+1),u] - q^*[s(t),u(t)]\} e_\lambda(s,u,t) \quad (1.17)$$

If we set λ to 0, we get $Q(0)$ which is nothing but the standard Q-learning algorithm we have described above.

- The other part of an actor/critic architecture is the *actor* which implements the policy. It follows a strict "law of effect" rule for action selection: actions that have been followed by a positive reward tend to be emitted more often while others tend to be emitted less often. But, it is the TD learning signal $r(t)+\gamma v^{\pi}[s(t+1)]-v^{\pi}[s(t)]$ generated by the critic to correct its predictions which is used by the actor to modulate its behavior.

The way an actor/critic architecture works is very intuitive. Suppose the actor switches to a new policy. If it is worse than the old one, the critic will over-predict the amount of reward collected, hence generating a negative TD learning signal that will punish the actor's behavior. On the other hand, if it is better than the old one, the critic will under-predict the amount of reward collected and hence, will generate a positive TD learning signal that will reward the actor's behavior.

The exact architecture of the actor is highly variable from one implementation to another and often depends on the characteristics of the task the agent has to master. So, actor/critic architectures do not completely solve the exploration/exploitation dilemma but they provide a very simple, elegant and efficient way to make the prediction and policy design stages interact.

1.3 Reinforcement learning and function approximation

1.3.1 The curse of dimensionality

Until now, we have supposed that the agent records the situation value or the action-situation value for each state and/or each control. So, in the limit of the convergence properties of the DP or NDP algorithm used, the exact situation or action-situation value for a state or a state-action pair can be known. This way of representing the optimal value function is called a *look-up table* representation because it is similar to have a table whose rows and columns are the states and actions while the values of the optimal value function are written in the cases.

But, such a kind of representation cannot be used if $|S|$ and/or $|U|$ are too large. This is the *curse of dimensionality.* For DP, even a single iteration of equation (1.10) would take too much time and so, conditions described in

equation (1.11) would never be met. For some applications, it would even be impossible to compute V_1^*. For NDP, some states will never be sufficiently visited and some actions will never be sufficiently emitted in some states for the behavior of the agent to be optimal. Up to a point, some states will never be visited at all and some actions will never be emitted at all.

To overcome the curse of dimensionality, the agent must be able to make inferences based on its current experience with the environment. This cannot be done with a look-up table representation of the optimal value function. It requires the use of a *function approximation architecture*.

A function approximation architecture A is able to approximate any function f having a specific functional form F between two sets X and Y if the values of a set of free parameters w are set appropriately. This is generally done through a *training stage* where a learning algorithm optimizes the approximation done by A on a subset $X_t \subset X$ (called the *training set*) according to a given performance criterion. After that training stage, A is able to approximate f for any $x \in X$ even if $x \notin X_t$.

The quality of these approximations is determined by the values of w_i which are themselves determined by the quality of the learning algorithm used to set them and by X_t. For instance, if the elements of X_t are not representative of X, then the value of the w_i will not be set appropriately to approximate f no matter the quality of the learning algorithm (see Bishop, 1995 for a deeper discussion of this topic). Another problem could arise if f has not the functional form approximated by A. For instance, if A only approximates linear functions and f is nonlinear, then, A will only be able to produce linear approximations of f.

Various function approximation architectures exist: decision trees, polynomial approximation,... We will concentrate here on one of the most popular architectures: *artificial neural networks* (ANN). These are programs whose functioning is roughly based on the one of the nervous system (Bishop, 1995; Haykins, 1999; McCulloch & Pitts, 1943). They have attracted a wide interest in recent years in cognitive science and artificial intelligence, because of their promises for technological applications and as models for brain and cognitive processes. They are the most appropriate function approximation architecture for RL and actually, the development of NDP is historically linked to ANN (see, Sutton & Barto, 1981).

1.3.2 Artificial neural networks

Figure 1.2 displays the schema of a standard artificial neuron, the basic processing unit in an ANN. Let's call this neuron neuron j. The algorithm determining the output of the neuron runs as follows:

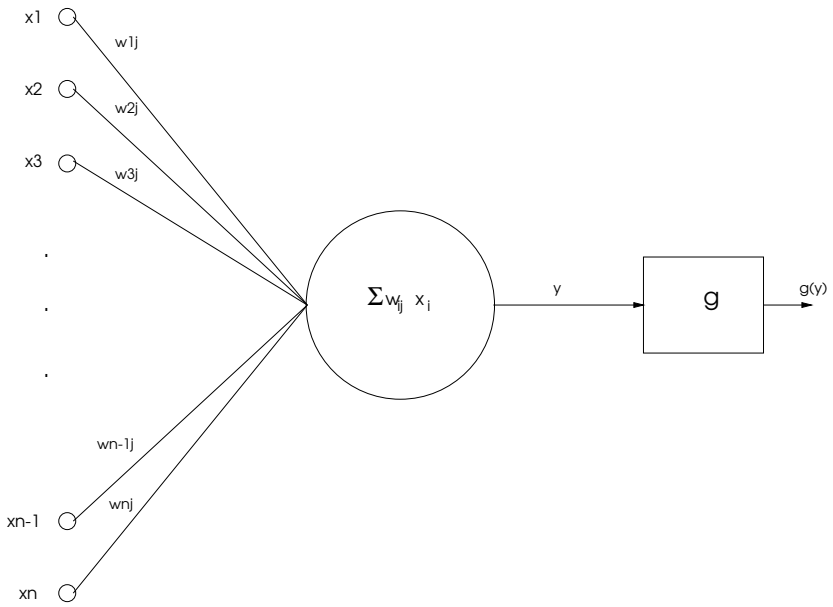

Figure 1.2: Schema of a standard artificial neuron

1. The neuron receives an input coded as a n-dimensional vector $x = [x_1, x_2, ...,$
   $x_i, ..., x_n]^T$. Each component x_i of the vector could be the output of a neuron i connected to neuron j. In such a case, the same algorithm which determines the output of neuron j determines the output of neuron i. They could also come from an *input neuron* i connected to neuron j. Input neurons are a bit different from other neurons since their output is directly set by an external stimulus.

2. To each neuron i is associated a *synaptic weight* w_{ij}. These weights are the components of a weight vector $w_j = [w_{1j}, w_{2j}, ..., w_{nj}]^T$.

3. The input vector x and the weight vector w determine the activation level a_j of neuron j. We have [1]

$$\begin{aligned} a_j &= w_0 + w.x \\ &= \sum_{i=0}^{n} w_i x_i \end{aligned} \tag{1.18}$$

[1]Usually, w_0 is not actually a weight but a bias fixed to a constant value and which cannot be changed by the learning algorithm. It can be fixed to 0. For this bias to have an impact on the activation level of the neuron, x_0 is fixed to 1.

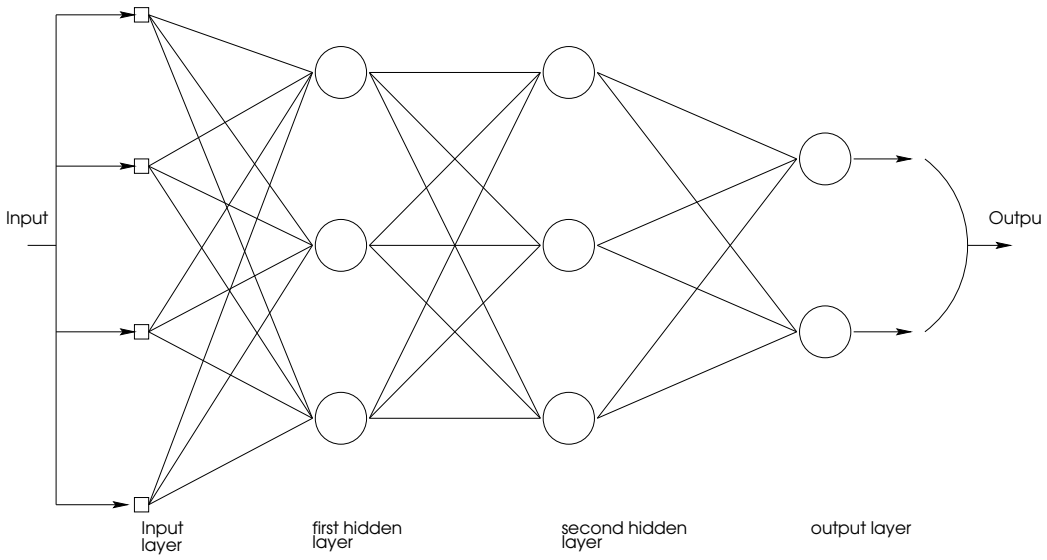

Figure 1.3: Schema of a feedforward network or perceptron

4. a_j is passed into an *activation function* g to produce the output y of the neuron. g can be linear, nonlinear or even stochastic, only determining the probability that the neuron changed its state, from active ($y = 1$) to inactive ($y = 0$) or vice versa. This output is then sent to other neurons or to the environment if the neuron is an *output neuron*.

Neurons like the one just described are connected to form networks. The way neurons are connected together defines the network architecture. The two most common architectures for function approximation are *radial basis function networks* and *feedforward networks* (Haykins, 1999; Bishop, 1995). We will only consider feedforward networks here since they are the most widely used.

Figure 1.3 shows the general organization of a feedforward network or *perceptron*. Its neurons are organized into *layers*: any neuron from layer $i-1$ is connected to any neuron from layer i but no neuron from layer i is connected to a neuron from layer $i-1$. If the network has only two layers, an *input layer* and an *output layer*, it is a *one-layered feedforward network* or a *simple perceptron*. If it has additional *hidden layers* between the input and the output layers, it is a *multilayered feedforward network* or a *multilayered perceptron*.

The first layer is the input layer. The activation levels of its neurons are set by a n-dimensional input vector. The activations of the input neurons are then used to determine the outputs of the neurons from the first hidden

layer (if there is such a layer) which are used to determine the output of the second hidden layer (if there is such a layer) and so on until the output layer is reached.

Suppose the input layer has n neurons and the output layer has m neurons. Then, the network can be seen as implementing a mapping from $\Re^n$ to $\Re^m$. To each n-dimensional input vector x whose components are the activation level of the n input neurons, the network associates a m-dimensional output vector whose components are the output of the m output neurons. The form of the mapping that a perceptron can implement depends on the activation function of its neurons and upon its architecture e.g. the number of layers and the number of neurons in a layer. For instance, a *linear perceptron*, e.g. a simple perceptron whose neurons have only linear activation functions, can only approximate linear functions (Minsky & Papert, 1969).

By adjusting the weights of the network with a learning algorithm, a perceptron can be used to approximate a specific function f. Suppose we want a perceptron with n input neurons and m output neurons to approximate a specific function $f : \Re^n \rightarrow \Re^m$ which associates to any $x \in X \subset \Re^n$ a m-dimensional vector $f(x) \in \Re^m$. We will also suppose that f can be appropriately represented by the neural network. Since X is a potentially infinite set, a finite training set $X_t \subset X$ is created. We have $|X_t| = N$ and the value of $f(x)$ for any $x \in X_t$ is known. To any $x \in X_t$, the network produces an output $o(x)$. The goal of the learning algorithm is to set the value of the weights in the network so as to minimize as much as possible the differences between $f(x)$ and $o(x)$ for all $x \in X_t$.

For this, we need a measure of the performance of the network, how good it is at approximating f on X_t. One of the most popular criteria is the *average squared error energy* ξ

$$\xi = \frac{1}{N} \sum_{x \in X_t} \xi(x) \tag{1.19}$$

with

$$\xi(x) = \frac{1}{2} \sum_{i=1}^{m} [f_i(x) - o_i(x)]^2 \tag{1.20}$$

where $f_i(x)$ and $o_i(x)$ are respectively the ith component of $f(x)$ and $o(x)$. It can be shown (Haykins, 1999) that a way to minimize ξ is to modify the weight w_{ij} between neuron i and neuron j each time an input vector $x \in X_t$ is presented with the *generalized delta rule*

$$w_{ij} := w_{ij} + \alpha \delta_j(x) y_i(x) \tag{1.21}$$

where α is a learning rate, $y_i(x)$ is the output of neuron i when vector x is presented at the input layer and $\delta_j(x)$ is the local error gradient for neuron

j e.g. a measure of the difference between its current output and the output it should have if $o(x) = f(x)$. If neuron j is an output neuron, then

$$\delta_j(x) = f_j(x) - y_j(x) \tag{1.22}$$

while, if neuron j is a hidden neuron (e.g. belongs to a hidden layer), we have

$$\delta_j(x) = g'[a_j(x)] \sum_{k \in K} w_{jk}\delta_k(x) \tag{1.23}$$

where g' is the first derivative of the activation function of neuron j, $a_j(x)$ is the activation level of neuron j when x is presented at the input layer and K is the set of neurons to which neuron j is connected in the next layer (e.g. either the next hidden layer or the output layer).

Equation (1.21), (1.22) and (1.23) form the basis of the popular *error backpropagation algorithm* (Haykins, 1999; Rumelhart, Hinton, & Williams, 1986) used to train weights in multilayered perceptrons: after $o(x)$ has been computed, the local error gradients are first computed for the output neurons and used to change the weights between the output neurons and the neurons from the last hidden layer. Knowing the local error gradients for the output neurons allows the computation of the local error gradients for the last hidden layer which allows the modification of the weights between this layer and the penultimate hidden layer and so on until the first hidden layer is reached. The triggering event is the ability to compute the local error gradients for the output neurons.

If the network is a simple perceptron, only equations (1.21) and (1.22) are necessary. If, moreover, the activation function of the neurons is linear, combining these two equations leads to

$$\begin{aligned} w_{ij} &:= w_{ij} + \alpha[f_i(x) - y_j(x)]x_i \\ &:= w_{ij} + \alpha \left[f_i(x) - \sum_{i=0}^{n} w_{ij}x_i \right] x_i \end{aligned} \tag{1.24}$$

which is known as the *delta rule,* the *LMS algorithm* or the *Widrow-Hoff rule* (Bishop, 1995; Haykins, 1999; Widrow & Hoff, 1960).

1.3.3 Neurodynamic programming and artificial neural networks

How can a feedforward network be used to evaluate the optimal value function? The function to be evaluated is either V^π, V^*, Q^π or Q^*. Let's take the example of evaluating V^π. Up to now, we do not care about the number

of layers of the network. It has n input neurons and a single output neuron which returns an approximation of V^π.

First, each state $s \in S$ must be encoded as a n-dimensional vector, which will then be used as an input for the network, through a *preprocessing stage* (Bertsekas & Tsitsiklis, 1996; Bishop, 1995). It can be defined as a mapping x with

$$\begin{aligned} x &: S \to \Re^n \\ &: s \mapsto x(s) = [x_1(s), x_2(s), ..., x_n(s)]^T \end{aligned} \tag{1.25}$$

This preprocessing stage is very important. First, it reduces the dimensionality of the input space, hence avoiding the curse of dimensionality. This reduction of the problem complexity also introduces correlations between the states which will be used by the network for generalization. Second, it is a way of introducing prior knowledge about the structure of the state space into the system. Various techniques have been developed (see Bishop, 1995 for a coverage of these various techniques as well as for a deeper discussion of the role and of the importance of input preprocessing in neural network training).

$x(s)$ is then used to produce an output $o(x)$. So, the network is implementing the function

$$\begin{aligned} o &: \Re^n \to \Re \\ &: x \mapsto o(x) \end{aligned} \tag{1.26}$$

More shortly, if we introduce the function $h = x \circ o$, the network can be considered as implementing

$$\begin{aligned} h &: S \to \Re \\ &: s \mapsto h(s) \end{aligned} \tag{1.27}$$

We will use the generalized delta rule to set the weights of the network so that $h(s)$ would be as close as possible to $V^\pi(s)$ for all s. To do this, we just need to be able to compute the local error gradient $\delta(s)$ for the output neuron i. If we apply equation (1.22), then the equation for $\delta(s)$ should be

$$\delta(s) = V^\pi(s) - h(s) \tag{1.28}$$

but we do not know the value of $V^\pi(s)$. So, instead of the real value $V^\pi(s)$, we must use an approximation $\psi(s)$ of $V^\pi(s)$. This is exactly what is done by NDP algorithms (see equation 1.12). For instance, $TD(0)$ (e.g. $TD(\lambda)$ with

$\lambda = 0$) was using $\psi[s(t)] = r(t) + \gamma v^{\pi}[s(t+1]$ as an estimation of $V^{\pi}[s(t)]$ (see equation 1.15). So, using this formula in the context of a feedforward network, equation (1.28) becomes

$$\delta[s(t)] = r(t) + \gamma h[s(t+1)] - h[s(t)] \tag{1.29}$$

This local error gradient for output neurons provides the basis for a new algorithm called *approximate* $TD(0)$ (Sutton & Barto, 1998) which can be used to set the weight of a feedforward network. For instance, if the network is a linear perceptron, combining equation (1.15) with equation (1.24) leads to

$$w_i := w_i + \alpha \left[r(t) + \gamma \sum_{i=0}^{n} w_i x_i[s(t+1)] - \sum_{i=0}^{n} w_i x_i[s(t)] \right] x_i[s(t)] \tag{1.30}$$

To get an approximate version of $TD(\lambda)$, we just need to add eligibility traces to the input neurons and to use them in the above equation instead of the activation level $x_i[s(t)]$. We have

$$w_i := w_i + \alpha \left[r(t) + \gamma \sum_{i=0}^{n} w_i x_i[s(t+1)] - \sum_{i=0}^{n} w_i x_i[s(t)] \right] e_{\lambda}(i,t) \tag{1.31}$$

The same reasoning would allow to have an approximate version of actually any NDP algorithm. Let's take, for instance, Q-learning. To get an approximate Q-learning algorithm, let's induce an order on U so that we get controls $u_1, u_2, ..., u_m$ with $|U| = m$. In this way, we can redefine the Q* function to ease its approximation by a perceptron. We have

$$\begin{aligned} Q^* &: S \rightarrow \Re^m \\ &: s \mapsto Q^*(s) = [Q^*(s,u_1), Q^*(s,u_2), ..., Q^*(s,u_m)]^T \end{aligned} \tag{1.32}$$

The perceptron we will use to approximate this function will have n output neurons $n_1, n_2, ...n_m$, each one trying to evaluate $Q^*(s,u_i)$ for a given $u_i \in U$. To make things simple, we will assume that output neuron i is trying to approximate $Q^*(s,u_i)$ for the ith neuron. We will also introduce the function $A(n_i,u_j)$ which is equal to 1 if output neuron i is keeping track of the Q*-value for control u_j and 0 if not. This perceptron can be considered as implementing the function

$$\begin{aligned} h &: S \rightarrow \Re^m \\ &: s \mapsto h(s) = [h_1(s), h_2(s), ..., h_m(s)]^T \end{aligned} \tag{1.33}$$

where $h_i(s)$ is the output of neuron n_i when state s is presented as an input. So, if the same reasoning as the one used to derive the local error gradient

for approximate $TD(\lambda)$ is used, we find that the error gradient $\delta[s(t), n_i]$ for each output neuron n_i used by the approximate Q-learning algorithm is

$$\delta[s(t), n_i] = \left\{ r(t) + \gamma \max_j h_j[s(t+1)] - h_j[s(t)] \right\} A[n_i, u(t)] \tag{1.34}$$

Because of $A[n_i, u(t)]$ term, only the connexions to the output neuron associated with the control just emitted are modified.

To summarize, the main differences between approximating the optimal value function with a feedforward network and standard function approximation with such networks are the following ones:

1. Evaluations of the optimal value function obtained through prompts of the environment by the agent are used to compute the error gradient at the network's output layer instead of the real values of the optimal value function which are, of course, unknown, since it is what the algorithm is trying to compute. Actually, the quality of the approximations made by a network after it has been trained on the training set X_t is usually evaluated by seeing how good it is at evaluating the function f on another subset $X_g \subset X$ in a generalization test (see Bishop, 1995 and Haykins, 1999). But, here, since the values of the optimal value function are unknown outside the network's approximation, it is impossible to test the quality of these approximations.

2. The training set X_t is unknown. Weights are updated by the learning algorithm on a real time basis as indicated by the introduction of time in equations (1.29) and (1.34).

Finally, note that a look-up table representation of the optimal value function could be considered as a special case of a neural network representation. Suppose that we induce an order on S like we did on U so that we have state $s_1, s_2, ..., s_i, ..., s_n$ with $|S| = n$. We will use the following preprocessing mapping

$$\begin{aligned} x &: S \to \Re^n \\ &: s_i \mapsto x(s) = [x_1(s_i), ..., x_j(s_i), ..., x_n(s_i)]^T \end{aligned} \tag{1.35}$$

with $x_j(s_i) = 1$ if $i = j$ and 0 otherwise. The reader can check for himself that if we use this preprocessing mapping with a linear perceptron designed to compute V^π and if we use equation (1.31) to set the weights of the network, then this approximate version of $TD(\lambda)$ is equivalent to the initial version (equation 1.15) and $w_i = v^\pi[s_i]$.

1.4 Conclusion

RL algorithms have become increasingly popular in artificial intelligence, mainly because of their success in applied settings (see, for instance, Crites & Barto, 1996; Dorigo & Columbetti, 1994 and Tessauro, 1994). Moreover, MDP bear striking similarities with the notion of contingencies we introduced at the beginning of this article. The elements of the organism's environment are the states of the MDP while the dependency relations (that is to say, the contingencies) are the transition probability $p(s'|s,u)$. Pavlovian contingencies in a MDP appear when, for any $u_1 \in U$ and $u_2 \in U$, $p(s'|s,u_1) = p(s'|s,u_2)$.

So, if any environment can described as a set of Pavlovian and operant contingencies, it is very tempting to assume that any set of Pavlovian and operant contingencies can be modelized by a MDP. Hence, if we assumed that behavior is the output of an optimization process (as many theorists in behavioral ecology, economics, evolutionary biology or psychology assume. See, for instance, Parker & Maynard Smith, 1990; Charnov, 1976 or Baum, 1981), this would mean that RL algorithms could at least be used to compute the behavior of an animal in a given situation. If this is possible, a more radical claim would be that animals are actually using some kind of RL algorithms to adapt their behavior to their environment. Indeed, the next section will review researches pointing out to striking similarities between adaptive mechanisms in animals and some NDP algorithms.

Chapter 2

Applications to conditioning

2.1 Conditioning: basic procedure and terminology

In the first part of this text, we had proposed an introduction to reinforcement learning and Markovian decision problems. We will now review their applications to conditioning. We begin by a review of the basic procedure and terminology in Pavlovian and operant conditioning since we will often used it in what follows.

2.1.1 Operant conditioning

Operant conditioning is a procedure developed by Skinner (1938). It is the main experimental protocol for the study of how organisms adjust their behavior to an operant contingency. It simply consists of the creation of an operant contingency and of the recording of its effect on a target behavior.

In the basic procedure, the operant contingency is created between a behavior and a consequence. This is the so-called *two-term contingency.* The behavior is called the *operant,* operant behaviors being the set of behaviors that can be modulated by their consequences. If the consequence increases the probability of emission of the operant, then it is a *reinforcer.* Making a reinforcer contingent to an operant is called *reinforcement.* The operant is then said to be *reinforced.* If, on the contrary, the consequence reduces the probability of emission of the operant, it is called a *punisher.* Making a punisher contingent to an operant is called *punishment.* The operant is then said to be *punished.*

Sometimes, a two-term contingency between an operant and a consequence is only effective when a given stimulus is present in the environment.

In such a case, we have a *three-term contingency.* If an animal learns that a given two-term contingency is only effective in the presence of a given stimulus (so that the probability of emission of the operant is increased or decreased only in the presence of that stimulus), then the stimulus has become a *discriminative stimulus.* The related training procedure is called a *discrimination.* More complex operant conditioning procedures are composed of sequences of three-term and two-term contingencies.

Discriminative stimuli predicting reinforcement can usually be used as reinforcers while discriminative stimuli predicting punishment can usually be used as punishers. Since these new functions of the stimuli are emergent products of the training procedure, they are said to be *conditioned* reinforcers and punishers. This contrasts with stimuli whose reinforcing or punishing properties seem to be innate rather then learned (such as food for an hungry animal, for instance) and which are said to be *primary* reinforcers and punishers.

Operant conditioning can be adapted to a wide range of species but, the vast majorities of the studies have been conducted, mainly for practical reasons, on rats and pigeons, placed in an experimental cage called a *Skinner box.* The animal is free to move in the box so that the number of manipulations of the animal by the experimenter is minimized. Moreover, electronic devices allow the automatic recording of responses and reinforcers. Operant responses are usually lever pressing for rats and key pecking for pigeons. Animals are hungry or thirsty so that food and water can be used as reinforcers. Electric shocks are the most widely used punishers. The effect of the consequence on behavior is evaluated through its impact on the rate of responding of the animal, e.g. the number of responses emitted per unit of time.

Since Skinner's early works in the thirties, a vast amount of data has been collected on the way animals adjust their behavior to their consequences (see Staddon & Honig, 1977; Hearst, 1988; Staddon, 1983 and Williams, 1988 for reviews). For instance, a wide range of studies has been devoted to *reinforcement schedules* e.g. the way the reinforcer is delivered as a function of responding. In *ratio schedules,* a certain number of responses (the *ratio* of the schedule) must have been emitted for the reinforcer to be delivered. In *fixed-ratio schedules* (FR), this number is fixed while in *variable-ratio schedules* (VR), it varies around a mean determined by the ratio of the schedule. In *interval schedules,* a minimal time lapse (the *interval* of the schedule) must have passed since the last reinforced response before a new response can be reinforced. In *fixed-interval schedules* (FI), the interval is fixed while it varies around a mean (determined by the interval of the schedule) for *variable-interval schedules* (VI). All those schedules have different and characteristic effects on behavior (Fester & Skinner, 1957; Williams, 1988; Zeiler,

1977). Performance under variable schedules is regular while it is not under fixed schedules. Moreover, response rate is higher under ratio schedules than under interval schedules.

2.1.2 Pavlovian conditioning

Just as operant conditioning is the main experimental procedure for the study of operant contingency, *Pavlovian conditioning* (Pavlov, 1927) is the main experimental procedure for the study of Pavlovian contingencies.

In a typical Pavlovian experiment, a *neutral stimulus* (NS), eliciting no specific reaction from the animal, is followed by an *unconditional stimulus* (US) which triggers an *unconditional response* (UR). For instance, in Pavlov's famous experiments with dogs, a tone (NS) was followed by the presentation of food powder in the mouth (US) that caused salivation (UR). The relation between the US and the UR is called an *unconditional reflex*. After several NS-US pairing, the NS alone triggers a response which is, most of the time, similar to the UR. So, in Pavlov's experiments, the dogs were salivating once they heard the tone. The NS is then no longer neutral and is called, from that point, a *conditional stimulus* (CS) while the response it now elicits is called a *conditional response* (CR). The relation between the CS and the CR is called a *conditional reflex*.

None modern study of Pavlovian conditioning is using dogs and food powder. Most of them are conducted with rats, pigeons and rabbits. (Hearst, 1988)

- The procedure most frequently used with rats is called a *conditioned suppression* procedure. The rat is first trained to press a lever for food under a VI (VI schedules can sustain a very high and regular rate of responding which can be used as a baseline to assess the effect of various experimental manipulations). A light (NS) is then followed by the delivery of an electric shock (US). After several pairings, the light causes a decrease in the rate of responding, supposed to be caused by fear (CR).

- In pigeons, the procedure used is called *autoshaping*. The pigeon is put in a Skinner box with a response key and a feeder on a wall. The response key is illuminated (NS) and this is followed by the delivery of food in the feeder (US). Being hungry, the pigeon pecks at the food (UR). After several pairings, the pigeon will peck at the illuminated response key (CR).

- The procedure used with rabbits is the *conditioning of the nictating membrane response* (NMR conditioning). The *nictating membrane* is a kind of second eyelid which does not exist anymore in humans. Animals like the rabbit can make it sweep over the eye by slightly retracting its eyeballs into the skull. This is the *nictating membrane response* (NMR). It is a UR triggered by US similar to the ones that would trigger an eye blink [1]. In NMR conditioning, the rabbit is immobilized, its eyes kept wide open by a machine while the movements of the nictating membrane are recorded. The NS is generally a tone or a light while the US is a puff of air. After several NS-US pairings, the NS alone is able to trigger the NMR.

- Finally, a few studies on humans are using tones as NS and eye blink or the galvanic skin response as UR [2].

2.2 Pavlovian conditioning and neurodynamic programming

2.2.1 The Rescorla-Wagner model

The learning process involved in Pavlovian conditioning was once seen as a very simple and elementary stimulus-response binding process. This is not the case any more. Pavlovian conditioning is now regarded as a procedure allowing an experimental investigation of the way animals learn about the causal structure of their environment, e.g. about the way animals learn what follows what and when (Wasserman & Miller, 1997). This is why, in most Pavlovian studies on humans, the subjects are simply exposed to successions of stimuli and are then asked to infer rules allowing them to predict what follows what (Schanks, 1994). The learning process is no more considered as simple but, on the contrary, as quite complex and sophisticated. One of the causes for this major shift in the opinion about Pavlovian conditioning could be found in the model proposed in 1972 by Robert Rescorla and Alan Wagner.

[1] Actually, in some studies, it is the eye blink response and not the NMR which is used as a UR. But, movements of the two eyelids are usually not independent while movements of the two nictating membranes are. So, the response of the second nictating membrane can be used as a control condition to assess the efficiency of conditioning.

[2] The galvanic skin response is a depolarization of the skin caused by emotionally loaded stimuli or by non-aversive very small voltage electric shocks

In the Rescorla and Wagner (1972)'s model, each CS[3] has an *associative value* $V(CS_i)$, supposed to be proportional to the amplitude of the CR or to the proportion of CRs triggered by the CS. A typical Pavlovian conditioning session is a succession of several trials, each trial being composed of the presentation of one or several CSs (or even, no CS) followed by the presentation (or the non-presentation) of the US. On each of these trials, the associative values of all the CS presented on a trial are updated according to the following equation

$$V(CS_i) := V(CS_i) + \alpha \left[r(t) - \sum_{i=1}^{n} V(CS_i) \right] \tag{2.1}$$

where n is the number of CS presented on that trial, $r(t)$ is the intensity of the US on that trial and α is a learning parameter.

Suppose there is only one CS. On each trial, its associative value is updated according to equation (2.1) until $V(CS) = r$: the associative value of a CS is a prediction of the intensity of the US. So, according to Rescorla and Wagner (1972), the animal tries to predict the intensity of the US based on the CSs present in the environment. These predictions are updated when they are not confirmed e.g. when the animal is surprised, e.g when $r(t) - \sum_{i=1}^{n} V(CS_i) \neq 0$.

Experimental supports

The Rescorla-Wagner (RW) model is certainly one of the most influential models of animal learning (Siegel & Allan, 1996). One reason for this popularity is its ability to synthesize numerous puzzling results about Pavlovian conditioning. Here are some examples.

In a *blocking* experiment (Kamin, 1969), a stimulus A is conditioned with a first group of subjects while another group receives no training. Then, both groups are conditioned with a compound stimulus AB composed of stimuli A and B. Finally, each component of the compound stimulus is presented separately to check if it triggers a CR. While in the second group, both A and B trigger it, only A does so in the first group. It is said that the prior conditioning of A has *blocked* the conditioning of B. According to equation (2.1), this happens because $V(A)$ has been set to r during the prior conditioning of A: this stimulus totally predicts the US. So, when A and B are

[3]The Pavlovian terminology, although coherent, is a bit confusing since a given stimulus is sometimes called a NS or a CS, depending on the behavioral reactions it elicits. Actually, researchers tend to be a bit loose on the vocabulary and call a NS a CS from the beginning, even if it fails to elicit a response at the end of the training. For clarity, we will sometimes do the same, like in this case.

Table 2.1: Evolution of the associative value of the stimuli according to the Rescorla-Wagner model during the first five trials of an analogue of Wagner et al (1968)'s first condition. For simplicity, we have assumed that AB trials alternate with AX trials and that the US always follows the presentation of AB. $\alpha = 0.25$.

	Trials				
	AB1	AX1	AB2	AX2	AB3
V(A)	0	0.25	0.187	0.328	0.262
V(B)	0		0.25		0.39
V(X)		0		-0.062	
US Intensity	1	0	1	0	1
Prediction	0	0.25	0.437	0.266	0.653
Error	1	-0.25	0.562	-0.266	0.348
New V(A)	0.25	0.187	0.328	0.262	0.348
New V(B)	0.25		0.39		0.477
New V(X)		-0.062		-0.129	
V(B)-V(A)	0		0.062		0.126

presented together, the increment in the associative value of B, $V(B)$, is $V(B) = V(B) + \alpha[r - V(A) + V(B)] = 0$ and so, the associative value of B never increases. A prediction that can be drawn from this analysis is that, if the intensity of the US is increased during the conditioning of AB, then A will no more fully predict the US . So, $r - V(A) + V(B)$ will be positive and so, B will be conditioned. This phenomenon, called *unblocking,* has been confirmed experimentally (Kamin, 1969).

In a typical Pavlovian procedure, $P(US|CS)$, the probability that the US is presented if the CS has been presented, is equal to 1 while $P(US|\overline{CS})$, the probability that the US is presented while the CS has not been presented, is equal to 0. Rescorla (1968) has manipulated this last probability, varying it from 0 to 1 while maintaining $P(US|CS)$ equal to 1. He discovered that the higher $P(US|\overline{CS})$, the less important the magnitude of the CR triggered by the CS was. In the extreme, when $P(US|\overline{CS})$=1, the CS did not trigger any CR at all. This is called a *contingency effect*. The explanation of this phenomenon by the RW model is similar to the one given for blocking, the role of the blocking stimulus A being played by the context of the experiment.

An experiment by Wagner, Logan, Haberlandt, and Price (1968) perfectly illustrates the sophistication of the learning process in Pavlovian conditioning. Wagner et al. (1968) used three stimuli A, B and X. In a first group,

Table 2.2: Evolution of the associative value of the stimuli according to the Rescorla-Wagner model during the first five trials of an analogue of Wagner et al (1968)'s second condition. As in Table 2.1, we have assumed that both kinds of trials alternate and that the US always follows the presentation of a compound stimulus. $\alpha = 0.25$.

	Trials				
	AB1	AX1	AB2	AX2	AB3
V(A)	0	0.25	0.437	0.516	0.59
V(B)	0		0.25		0.33
V(X)		0		-0.187	
US Intensity	1	1	1	1	1
Prediction	0	0.25	0.688	0.703	0.92
Error	1	0.75	0.312	0.297	0.08
New V(A)	0.25	0.437	0.516	0.59	0.61
New V(B)	0.25		0.33		0.35
New V(X)		0.187		0.262	

the compound stimulus AB was followed by the US half of the time while it was never the case for AX. So, for this group, AB predicted the US while AX did not but B by itself was a sufficient predictor. In this group, only B triggered a CR. In another group, the US followed both AB and AX in 50 per cent of the trials. All the stimuli predicted the US with equal efficiency but it was less costly to keep track only of the presence or absence of A. In this group, only A triggered a CR. These results are also explained by the RW model. Consider Wagner et al. (1968)'s first condition where AB is followed by the US but not AX. Table 2.1 shows how the associative values of the stimuli evolve from trial to trial. The discrepancy between the value of B and the value of A increases while the prediction error in AB trials decreases. So, in the long run, B will fully predict the US and will block further conditioning of A. The explanation for Wagner et al. (1968)'s second condition is presented in Table 2.2. The value of A rapidly increases and is much higher than the values of B and X. So, the conditioning of A is better and, in certain conditions, it could even be the only stimulus triggering a CR.

In *superconditioning* (Rescorla, 1971), a stimulus A is trained to be a signal for the *non-presentation* of the US [4]. Then a compound stimulus AB is

[4]Such a stimulus is a *conditioned inhibitor*. This is done by conditioning a stimulus C

conditioned. Presented alone, B triggers a stronger CR than in other groups where it has been conditioned alone or with another stimulus not trained as a conditioned inhibitor. According to the RW model, this is because $V(A)$ is negative and so $V(B) := r - V(A) - V(B) = r + |V(A)| - V(B)$ which is, of course, greater than $r - V(B)$.

In *over-expectation* (Kremer, 1978), two stimuli A and B are trained separately with the same US. Then, the compound AB is conditioned with the same US as the one used to train A and B. Tested alone, both CSs trigger a CR less important than the one they triggered initially. The explanation is straightforward. Before the compound conditioning, we had $V(A) = V(B) = r$ and so, $V(AB) = V(A) + V(B) = 2r$. The US is clearly over-predicted and the compound conditioning will reduce the associative value of both A and B to $\frac{r}{2}$ so that $V(AB) = r$.

In addition to these phenomena, the RW model has been used to explain data outside the field of Pavlovian conditioning (from category learning to human causality judgements through interpersonal attraction. See Siegel & Allan, 1996). Although it has not been unchallenged (Miller, Barnett, & Grahame, 1995), it remains highly influential within the field of animal learning and a lot of more recent and elaborated models are no more than developments of the original RW model (see below, TD learning).

2.2.2 Neurodynamic programming and the Rescorla-Wagner model

The TD model

Comparing equation (2.1) with equation (1.12) reveals that both of them are describing the same learning process: in both cases, a prediction of reward is compared to a better evaluation and the discrepancy between both is used to update the reward prediction. But the analogy goes further.

The RW model can be implemented by a linear perceptron (Sutton & Barto, 1981). This network is displayed in Figure 2.1. S, the state space of the environment in which this network is embedded, is composed of all the possible combinations of CSs that the network can encounter from one

with a US and then omitting it when presenting the compound stimulus AC. The animal learns to omit the CR on AC trials. Moreover, if a stimulus D has been conditioned to the US, the CR will also be omitted if the compound AD is presented. Conditioned inhibition is easily explained by the RW model but it makes the false prediction that a conditioned inhibitor will extinguish if it is presented alone. This is not confirmed experimentally. To account for that, more advanced versions of the RW model (including the TD model presented below) suppose that $\sum_{i=1}^{n} V(CS_i)$, the prediction of the US, cannot be negative. Its lowest possible value is 0.

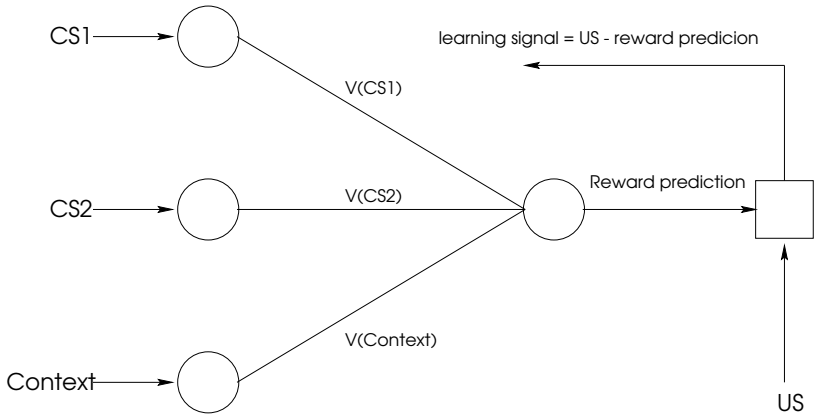

Figure 2.1: A neural network implementation of the Rescorla-Wagner model

trial to another. So, for instance, if the network is used to modelize data from an experiment where there are only two CS (A and B), then $S = [A, B, (A, B), \emptyset]$. Each input neuron of the network codes the presence or absence of a specific CS on a given trial: $x_i[s(t)]$, the activation level of the ith input neuron on trial t where the configuration $s(t)$ is presented to the network is 1 if CS_i is present during that trial, 0 otherwise. The weights of the network code the associative values of the CSs so that w_i, the strength of the synaptic connection between the input neuron i and the output neuron, is equal to $V(CS_i)$. The network's input vector is updated at every trial. In that context, the RW equation becomes

$$w_i := w_i + \alpha \left\{ r(t) - \sum_{i=1}^{n} w_i x_i[s(t)] \right\} x_i[s(t)] \tag{2.2}$$

This is exactly equation (1.30) with $\gamma = 0$. So, the RW model of Pavlovian conditioning is nothing more but approximate $TD(0)$ applied to a linear perceptron with $\gamma = 0$!

This is not surprising. The first works on NDP (Sutton & Barto, 1981) explicitly took their inspiration from the psychology of animal learning and, more specifically, from the RW model. When the first versions of $TD(\lambda)$ were introduced, they were presented as real-time extensions of the RW model[5] (Sutton & Barto, 1981). Using $TD(\lambda)$ in combination with a linear perceptron is actually known in the field of Pavlovian learning as the *TD model* of Pavlovian conditioning (Sutton & Barto, 1990). It is considered as one of the

[5]For another real-time extension of the RW model using differential equations and used in several models of Pavlovian and operant conditioning, see Schmajuk (1997)

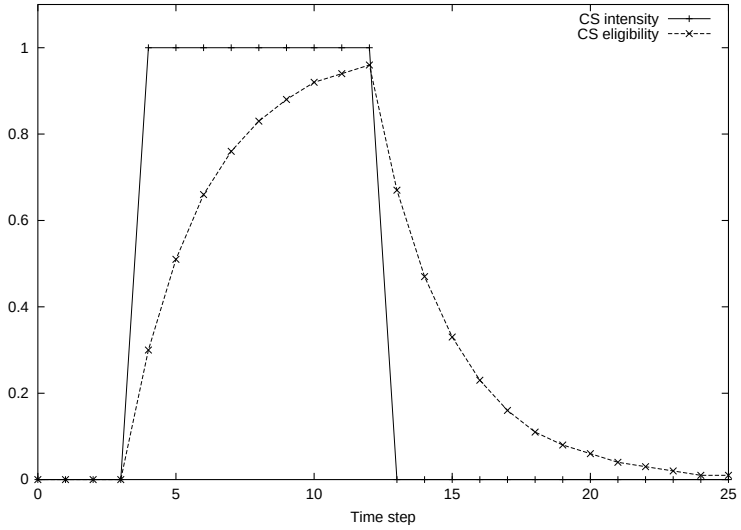

Figure 2.2: Evolution of the eligibility trace for a CS in the TD model

most successful theories of learning in a Pavlovian context (Moore & Choi, 1997).

The TD model is a one-layered network just like the one depicted in Figure 2.1. The state space S for the environment in which the TD network is embedded is defined in the same way as for this network. The only differences are that each input neuron of the TD network codes the presence or absence of a given CS at a given time t (and no more on a given trial t) and that approximate $TD(\lambda)$ algorithm for linear perceptrons (equation 1.31) is used to set its weights instead of equation (2.2). Being a sophisticated version of the RW model, the TD model is able to account for all the phenomena explained by the original RW model (blocking, prediction sufficiency,... see Sutton and Barto, 1981, 1990) but, because it is a real-time model, it can go beyond these phenomena observed on a trial level to account for intratrial ones. To do this, it uses a special kind of eligibility traces whose temporal evolution is controlled by the following equation:

$$e_\lambda(i,t) = e_\lambda(i,t-1) + (1-\lambda)x_i[s(t)] - (1-\lambda)e_\lambda(i,t-1) \tag{2.3}$$

Figure 2.2 shows how this eligibility trace evolves as a function of CS duration.

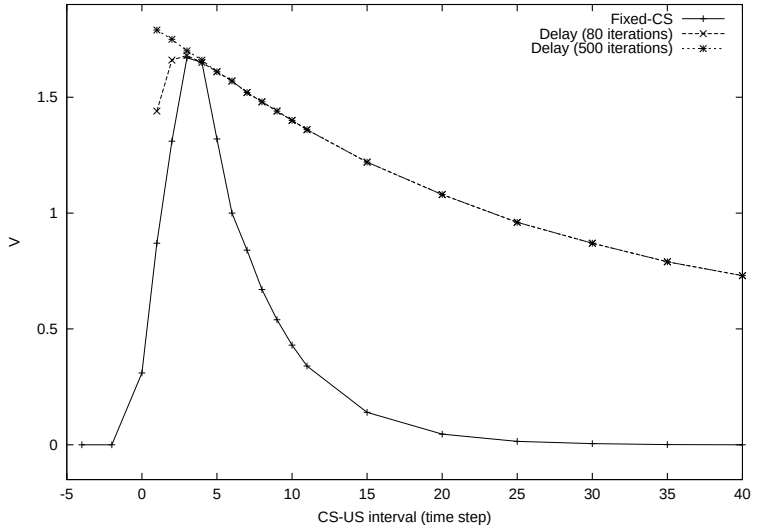

Figure 2.3: ISI curves generated by the TD model using a 4-time step CS and a 2-time step US. If the duration of a time step is fixed to 50 ms, these correspond to stimuli used in NMR conditioning experiments. While in a fixed-CS procedure, the duration of the CS is fixed, it only stops at the time of US onset in a delay procedure.

Simulations of real-time features of Pavlovian conditioning

Here are some instances of intratrial phenomena explained by the TD model (Sutton & Barto, 1990).

One important variable in Pavlovian conditioning is the *interstimulus interval* (ISI) e.g. the interval between the onset of the CS and the onset of the US. The function relating the ISI to the efficiency of conditioning (measured as the proportion of CR generated by a CS) is an inverted U-shaped function. It is null for negative or null ISI (the US begins before or just at the same time as the CS), it then increases until an optimal ISI is reached. From that point, it decreases and falls back to 0. This inverted U-shaped function is found no matter the duration of the CS, the intensity of the US, the kind of US or the intertrial interval although all these variables have a significant impact on the absolute value of the ISI for optimal conditioning (Kehoe, 1990). As Figure 2.3 shows, if the maximal US prediction generated in the presence of a CS is considered to be proportional to the amount of CR generated by that CS, the TD model reproduces the inverted U-shaped ISI curves. It correctly predicts that the tail of the ISI curve observed in a delay-CS procedure is longer than the one of the curve observed in a fixed-

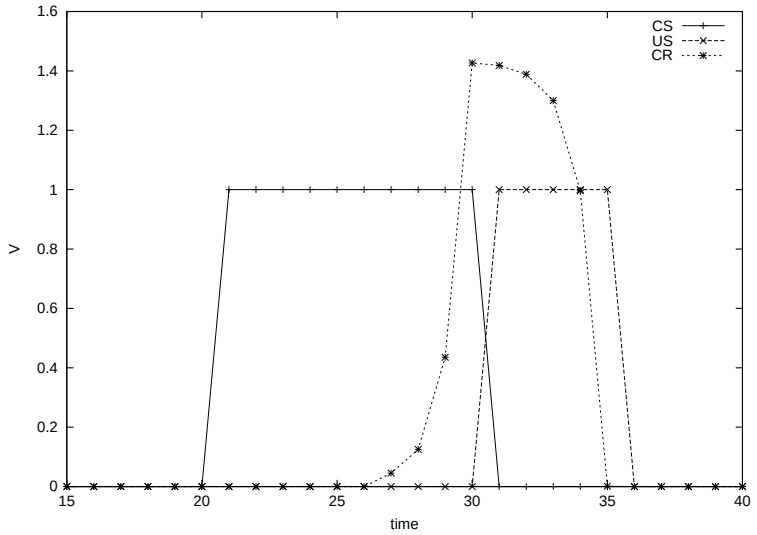

Figure 2.4: CR topography generated by the TD model if a tapped delay line is added to the neuron perceiving the presence of the CS

CS procedure. These are not predictions made by all models (see Sutton & Barto, 1990).

In trace conditioning, there is a delay between the offset of the CS and the onset of the US. Conditioning is better if that delay is filled by another stimulus (Kehoe, 1990). This phenomenon is simulated by the TD model (Sutton & Barto, 1990).

If a stimulus A begins before another stimulus B and both of them finish at the same time, then A will be much better conditioned to the US than B (Kehoe, 1990). The TD model not only accounts for this phenomenon but also makes the further prediction that this will happen even if B has previously been conditioned to the US. This prediction was validated by Kehoe, Schreurs, and Graham (1987).

A stimulus can also become a CS if it is paired with an already established CS. Again, this is a feature of Pavlovian conditioning captured by the TD model.

Simulations of the topography and timing of the conditional response

The TD model is also able to account for key features of CR topography and timing in NMR conditioning (Moore & Choi, 1997; Moore, Choi, & Brunzell, 1998). For this, it must be assumed that the temporal evolution of the CS is

represented at the input layer of the network by a *tapped delay line.* This is a common tool in neural network modeling practices for the representation of temporal events (Haykins, 1999). In a neural network using a tapped delay line, *m lag units* are associated with each primary input neuron whose activation levels $x_i(t)$ are directly set by the input vector coming from the environment. $x_i^j(t)$, the activation of the jth lag unit associated with the ith input neuron, is $x_i^{j-1}(t-1)$, $x_i^0(t)$ being $x_i(t)$. In that context, the TD model prediction for the amount of reward collected is

$$h[s(t)] = \sum_{i=1}^{n} \left\{ w_i x_i[s(t)] + \sum_{j=1}^{m} w_i^j x_i^j(t) \right\} \tag{2.4}$$

n is the number of primary input neurons and w_i^j is the weight of the connection between the jth lag unit associated with the ith input neuron. These weights are updated by the TD equation just like other weights. Actually, if not for the way their activation level is set, lag units behave exactly as regular input neurons. Indeed, each lag unit has its own eligibility trace.

Figure 2.4 shows a CR topography generated by the TD model with a tapped delay line when a linear relation is assumed between the predictions of the TD model and the degree of closing of the nictating membrane. These simulated CRs reach their maximum amplitude just before the onset of the US. This is a key feature of real CRs in NMR conditioning (Kehoe, 1990) and a robust characteristic of CRs generated by the TD model (Moore & Choi, 1997) although it can disappear if the parameters of the model are fixed to some specific values.

Complexifying a bit more the CS representation at the input layer, Moore and Choi (1997) showed that the TD model could account for further features of CR timing. They assumed that two other input neurons are associated to a given CS: one is activated by CS onset while the other is activated by CS offset. A tapped delay line is associated to each of these neurons. With these modifications, the TD model accounts for CR timing observed by Desmond and Moore (1991a) and Millenson, Kehoe, and Gormezano (1977) although it has some problem with the CR topography.

- Desmond and Moore (1991a) exposed rabbits to a 150 ms CS whose offset was followed by the US 200 ms later. Once conditioning was established, they observed that a 150 ms CS triggered a single CR pecking at 350 ms while a 500 ms CS elicited 2 CRs, one pecking at 350 ms and the other one at 550 ms. Desmond and Moore (1991a) interpret their results by assuming that CS onset and CS offset are two separate CSs. With a 150 ms, the CR triggered by CS onset is

synchronized with the one triggered by CS offset and so, only one CR is observed, while this is no more the case with a 500 ms CS and so, two CRs are generated. Moore and Choi (1997) showed that the TD model is able to account for these data. It also correctly predicts that the peak of the CR generated by the 150 ms CS is greater than the amplitude of the two CRs generated by the 500 ms CS. But, with a 500 ms CS, the peak of the second CR is lower than the peak of the first one while the TD model predicts that both CRs have the same amplitude.

- Using a delay-CS procedure (see the caption of Figure 2.3), Millenson et al. (1977) made a US follow CS onset by either 200 ms or 700 ms, both delays randomly alternating from one trial to the other. Then, they presented to their rabbits a 200 ms CS or a 700 ms CS. The 200 ms CS generated a CR peaking at 200 ms while the 700 ms CS generated two CRs, one peaking at 200 ms and the other one at 700 ms. Once more, Moore and Choi (1997) showed that this result was simulated by the TD model. But, if the CR timing was well captured, there was again a problem with the topographies. The TD model predicts that the second CR with a 700 ms has a higher peak than the first one while Millenson et al. (1977) observed that the amplitudes of the two CRs were similar.

Finally, Moore et al. (1998) conditioned rabbits with a 800 ms CS whose onset was followed by the US 300, 500 or 700 ms later. Intervals for US onset randomly varied from one trial to the next one. They observed two response strategies in their subjects. In what they called a *failsafe strategy,* the rabbit seemed to follow the rule "close eye quickly and keep it closed until the probability of US is minimal" while in the *conditional expectation strategy,* the rule was "close eye progressively as the conditioned probability of the US increases to maximum". With appropriate free parameter setting, the failsafe strategy was generated by the TD model and so was the conditional expectation strategy but only at the cost of a further complexification of the CS representation by the addition of what Moore et al. (1998) called a *marking process.* It would be too long to explain this new mechanism here. The idea is that each lag unit has its own tapped delay line but its ability to activate it is not fixed but can be changed through experience. See Moore et al. (1998) for further details.

Current limits of the TD model

So, at a behavioral level, the TD model is able to deal successfully with a wide range of data. Moreover, we will see that its biological plausibility goes even further since recent evidences are showing how it could be implemented in the brain. But, of course, it is not perfect. To conclude this part on the TD model, we would like to point out to two major defficiencies of current implementations of the TD model.

The first problem is actually shared by many other models of conditioning and is illustrated by a procedure called *sensory preconditioning* (Hall, 2002). In the first stage of this procedure, a neutral stimulus S1 is followed by another neutral stimulus S2. In the second stage, S2 is conditioned with a standard US. Finally, in the test stage, S1 and S2 are presented alone. As expected, S2 triggers a CR but *this is also the case for S1.* Obviously, even if the first stage of the experiment has not induced any behavioral change, the subject has learned something. A related experiment by Matzel, Held, and Miller (1988) is even more spectacular. After the initial exposure to the S1-S2 contingency, S2 is paired with a US according to a backward conditioning procedure: the US is *followed* by S2. This is known to led to very poor conditionng (check the ISI curve for negative ISI on Figure 2.3). When presented alone, S2 triggers no CR *but S1 does !*

Althought these results perfectly fit within a prediction framework, the TD model cannot account for them because it relies on two fundamental assumptions:

- Animals try to predict *rewarding or punishing events.* That is to say, they can learn Pavlovian contingencies only if one of the stimuli involved in the contingency is a US or an already trained CS. But sensory preconditioning clearly shows that animals are able to learn any kind of Pavlovian contingencies.

- Animals can learn a Pavlovian contingency between two stimuli only if they have been explicitely exposed to it. Again, these experiments seem to indicate that animals are able to "infer" contingencies that they have never directly experienced.

Future development in the TD model should remedy to these defficiencies[6]. Maybe this could be done by using algorithms combining features of DP and

[6]The sensory subnetwork in the model by Donahoe, Burgos, and Palmer (1993) seems to be a first step in that direction althought it remains unable to account phenomena as the one described in the study by Matzel et al. (1988) as well as other instances of "inferential" learning.

NDP (Sutton & Barto, 1998)[7].

The second problem has to do with the tapped delay line which is used in recent versions of the TD model. As we saw above, this device allows the TD model to account for CR timing but nobody seems to have noticed that *such a model is unable to account for the ISI effect.* This is because only the last lag units before US onset are conditioned. By manipulating the CS onset-US onset interval, different lag units are conditioned and this is why the CR is timed to US onset. But this also means that *the interval between the conditioned lag units and the US is always the same, no matter the CS onset-US onset interval.* Hence, there is no ISI effect in this model. Another way to represent time in the TD model is necessary to overcome this major problem.

2.2.3 Operant conditioning and the Rescorla-Wagner model

It is surprising that the first works on NDP took their inspiration from Pavlovian conditioning (Sutton & Barto, 1981) while NDP was designed to solve MDP which bears striking similarities with the tasks faced by an animal in an operant conditioning situation. It is even more surprising now that the MDP framework is well established among RL scholars.

Now, a growing body of researchers is considering that the learning process is the same in Pavlovian and operant conditioning (see, for instance, Donahoe et al., 1993) and so, if NDP is a good model of Pavlovian performance, it should also be a good model of operant performance. But there are few direct evidences of a predictive mechanism like the one described by the RW model and NDP in operant conditioning. This is because most of the procedures which support the RW model in the context of Pavlovian conditioning are difficult to transpose within an operant context.

Take blocking for instance. The operant analogue of a blocking result would be to show that a response is not reinforced if it is emitted at the same time as another response which already predicts the reinforcer. This raises two problems. First, responses cannot really be emitted simultaneously. Second, the experimenter must find a way to manipulate the animal's behavior to force it to emit the blocked and blocking responses. This kind of manipulation poses methodological problems and opens the door for alternative explanations. This is why demonstrations of blocking in an operant context have not been convincing (see, for instance, Williams, 1975 and the

[7]These algorithms (such as Sutton's Dyna architecture) are using NDP to build a model of their environment which allows them to use DP to compute Bellman's equation.

criticism by Zanich & Fowler, 1978).

The same problems apply to most of the phenomena explained by the RW model with the exception of the contingency effect. Just like the delivery of the US in the absence of the CS reduces the efficiency of conditioning, the delivery of noncontingent reinforcers reduces the rate of emission of the operant (Hammond, 1980). This could be caused by the reinforcement of other responses by the noncontingent reinforcers. These responses would then compete with the operant and hence, reduce its rate of emission. To test this hypothesis versus a RW type of explanation, Dickinson and Mulatero (1989) trained two operant responses in rats. Each response was reinforced by a different kind of reinforcer (either food pellets or a sugar solution). Then, they delivered noncontingent reinforcers of one type. They observed that only the rate of emission of the operant response associated with the reinforcer delivered noncontingently was reduced. This is incompatible with a response competition account of the contingency effect in operant conditioning since such an hypothesis predicts that the ratio of emission of both operant responses should decrease.

Moreover, the RW model explains these results by the fact that the context has become a predictor of the reinforcers. So, if we could reduce the predictive value of the context, the impact of the noncontingent reinforcers on the operant response should be reduced. This could be done by signaling with a stimulus all the noncontingent reinforcers. This was done by Hammond and Weinberg (1984) and Dickinson and Charnock (1985). They observed that the decrease in the rate of emission of the operant response is less important when the noncontingent reinforcers are signalled by a light.

These experiments are among the few ones which seem to demonstrate that a predictive learning mechanism is underlying operant conditioning. Another line of evidences comes from neurophysiological studies which have been looking for NDP in the brain and which were using operant conditioning procedures.

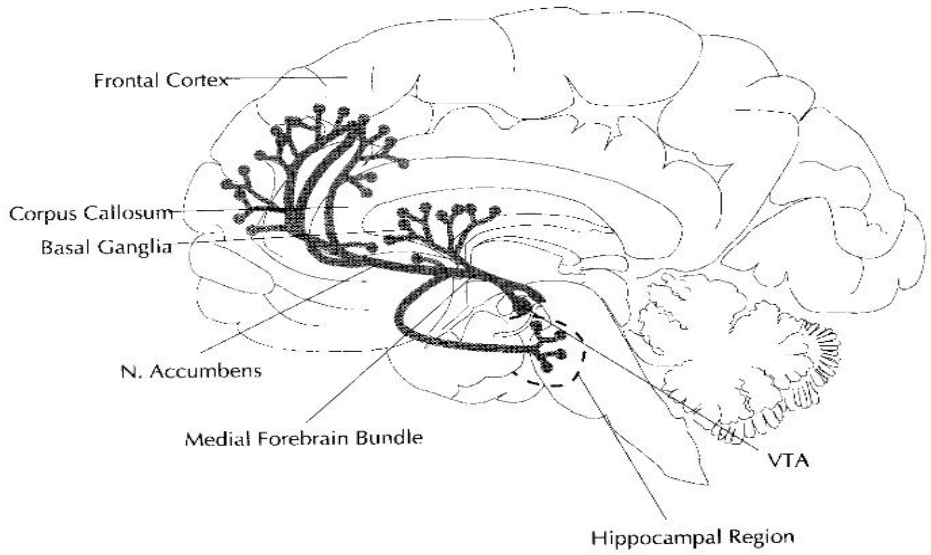

Figure 2.5: Localization of the main dopamine pathways in the brain (from Donahoe and Palmer, 1994)

2.3 Neurophysiological studies

2.3.1 Neurodynamic programming and the basal ganglia

The activity of midbrain dopamine neurons during learning in the monkey

Figure 2.5 shows the localization of the main *dopamine pathways* in the human brain. Their sources are the *substantia nigra* (SN) and the *ventro-tegmental area* (VTA), both located in the midbrain. From there, they widely project through the brain, especially to the *orbito-frontal cortex* and the *ventral striatum*. Both structures send projections back to the VTA and SN.

These pathways play a critical role in reinforcement (see Robbins & Everitt, 1996; Wise, 1996 and Wise & Rompre, 1989 for reviews). So, a rat will quickly learn to press a lever if this causes the electrical stimulation of brain areas close to dopamine sites, the most effective areas being the VTA, the SN and the *medial forebrain bundle* containing the ascending and descending fibers connecting the cortex to the VTA. These brain stimulations are actually so reinforcing that rats, and even more sophisticated animals like

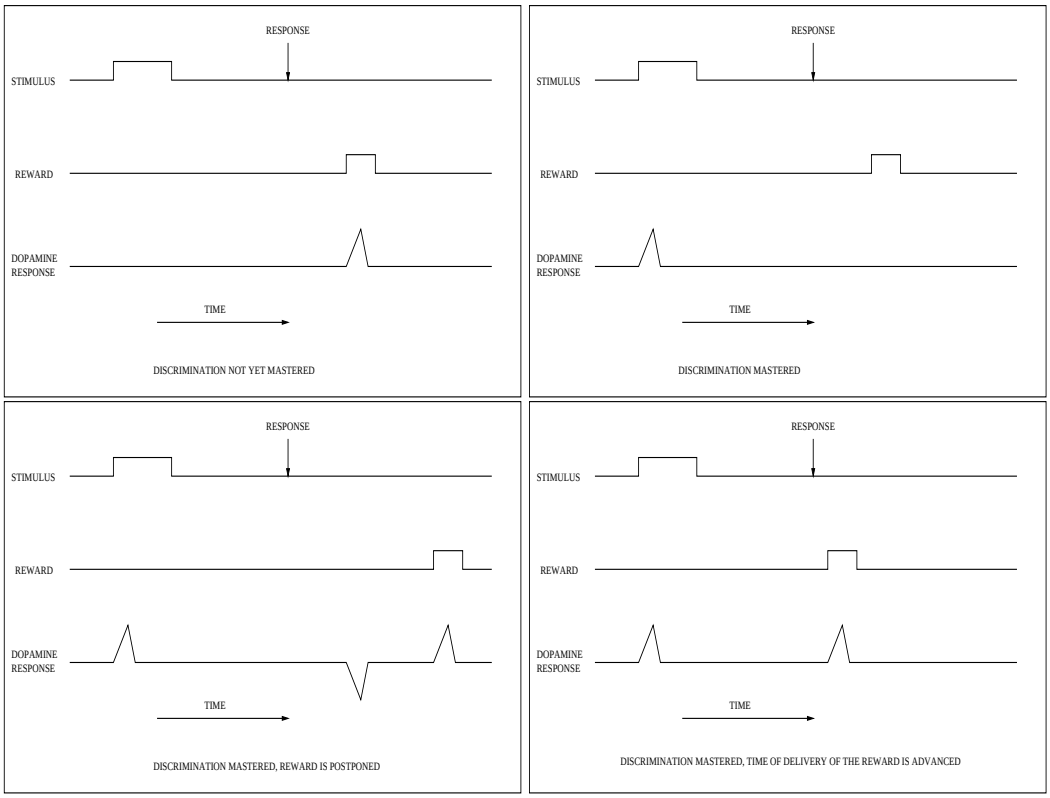

Figure 2.6: Response of monkeys' dopamine neurons at various stages of learning (adapted from results reported in Schultz et al, 1998). The line labeled "dopamine response" shows the changes in the dopamine neuron impulse frequency.

monkeys, would stop doing anything else (including eating) than pressing the lever (Robbins & Everitt, 1996). It is also demonstrated that the addictive effect of drugs like cocaine and amphetamine is mediated through their action on the metabolism of dopamine, especially at the level of the ventral striatum (Robbins & Everitt, 1996; Wise, 1996). So, for instance, a rat ceases to press a lever for cocaine self-injection or food after it had been given a dopamine antagonist at the level of the ventral striatum.

Another line of evidences comes from the laboratory of Wolfram Schultz and his colleagues (see Schultz, 1998 and Schultz, Tremblay, & Hollerman, 1998 for reviews) who have recorded the in vivo activity of dopamine neurons in monkeys. We will particularly review the results of two studies. In the first one (Romo & Schultz, 1990), monkeys reached into a food box in front of them where they could sometimes find food while, in the second one (Ljungberg, Apicella, & Schultz, 1992), apple juice, first freely delivered to

the monkey, was used to reinforce lever pressing in the presence of a stimulus (either a light or a sound). The main results from these two studies are summarized below and in Figure 2.6:

- The dopamine neurons were displaying a kind of constant background activity consisting of negative or positive impulses of rather long durations (1.8 to 5.5 ms) and low frequencies (0.5 to 0.5 imp/s).

- This background activity was increased for about 300 ms by certain stimuli like the delivery of apple juice into the mouth, the discovery of food in the food box after self-initiated movements and the onset of a novel stimulus or of a discriminative stimulus predicting reinforcement. By contrast, touching nonedible objects in the experimental box, arm movements or aversive stimuli like an air puff or a stimulus used in an avoidance task as a warning signal for the upcoming of an aversive event did not have that effect. The dopamine response was massive, involving about 80 per cent of the neuron recorded. It did not discriminate between reinforcers (apple juice versus food, conditioned versus unconditioned) or sensory modality (light versus sound), the same neurons responding to any of these reinforcing events.

- A reward failed to elicit a dopamine response if it was preceeded by a reliable reinforcer. So, in the discrimination task, the dopamine neurons did not respond any more to the delivery of the reward once the task was mastered. But, they reacted again if the time of delivery of the reward was advanced or delayed[8]. Moreover, in this last case, the dopamine neuron activity was depressed exactly at the usual time of delivery of the reward. This property of the dopamine response was also observed with conditioned reinforcers if their onset was predicted by another stimulus.

So, dopamine neurons seem to respond to the unpredicted onset of a conditioned or unconditioned reinforcer, no matter its sensory modality, by an increased activity and to the unpredicted nondelivery of a predicted reinforcer by a decrease activity.

These first results were confirmed by other studies using more complex tasks (Hollerman & Schultz, 1998; Schultz, Apicella, & Ljungberg, 1993) and where Schultz et coll. studied the evolution of the dopamine response during the course of learning. For instance, in a study by Hollerman and Schultz (1998), the monkey faced two pictures in front of it with a lever under each one. Pressing the lever under one picture was reinforced by apple

[8]The reward was delivered 5 seconds after the response

juice while pressing the lever under the other one was not. Once this task was mastered on a stimulus pair, another one was introduced, hence allowing the comparison between the dopamine neuron activity during learned and unlearned trials. Hollerman and Schultz (1998) observed that the loss in the dopamine response to the reward was gradual and correlated to the learning curve of the animal. Moreover, the dopamine response transferred to the onset of the pictures and was depressed at the usual time of delivery of the reinforcer when a monkey made an error on a learned trial. Schultz et al. (1993) got the same results with spatial discrimination tasks.

This pattern of activity contrasts with the one recorded by Schultz and coll in the ventral striatum (Tremblay, Hollerman, & Schultz, 1998) and the orbito-frontal cortex (Schultz et al., 1998). Tremblay et al. (1998) trained successive discriminations, each discrimination being composed of three stimuli: pressing a lever was reinforced by apple juice with one stimulus, by a sound with another one while doing nothing with the third one was also reinforced by apple juice. Once the task was mastered with a stimulus trial, another one was introduced and so on until all the scheduled discriminations were mastered. Tremblay et al. (1998) observed that neuronal firing in the striatum was related to the components of the task: a neuron fired either before or after a specific intratrial event (onset of the discriminative stimulus, onset of the signal for responding, delivery of the reinforcer) in a trial for a given kind of trial (movement reinforced versus no movement reinforced, apple juice used as reinforcer versus sound used as reinforcer), some neurons sometimes responding to two events and/or to the same event in two kinds of trials. For Tremblay et al. (1998), these neuronal patterns of activation reflected reaction to (activation after an event) and expectation of (activation before an event) the various task components. They were not built-in templates since they displayed modifications during the course of learning with some neurons sometimes changing their relationship to a given intratrial event or to a given kind of trial. In the same way, activity in the orbito-frontal cortex occurs before the delivery of reward and discriminates the kind of reward upcoming (Schultz et al., 1998).

For Schulz et coll. (Schultz, 1998; Schultz, Dayan, & Montague, 1997), these results indicate that the activity of dopamine neurons is coding an error in the prediction of the amount and time of occurrence of reward. More specifically, they propose that dopamine neurons are implementing the TD learning signal $r(t)+\gamma v^{\pi}[s(t+1)]-v^{\pi}[s(t)]$ used in $TD(\lambda)$[9]. This signal could be broadcast to the ventral striatum and the orbito-frontal cortex to induce

[9]Note that this implies an assumption incompatible with the TD model of Pavlovian conditioning. A negative TD learning signal is necessary to account for the fact that dopamine activity is depressed when a reward is omitted. But, RW-like models of condi-

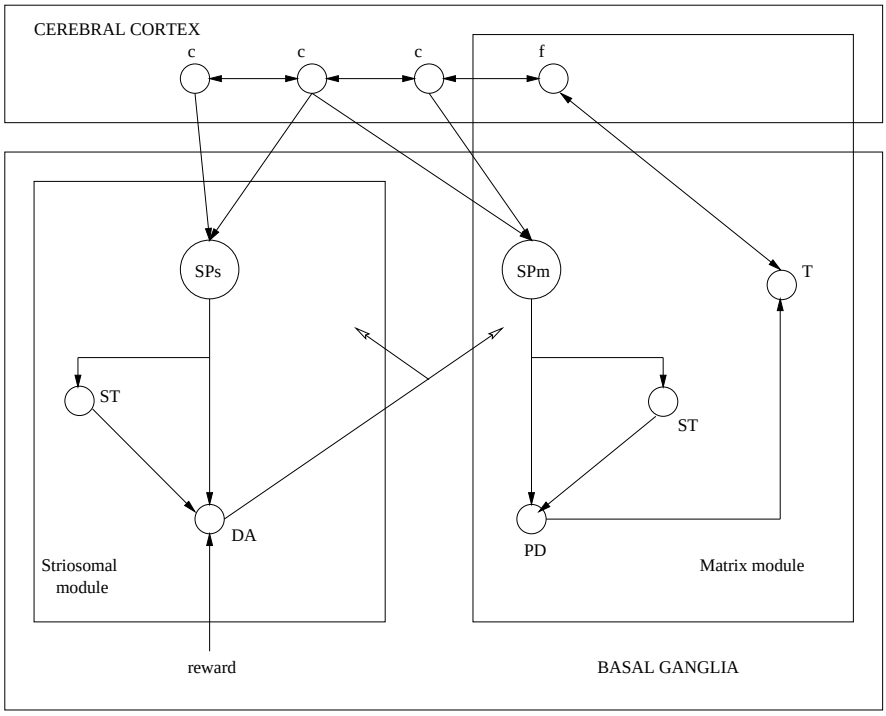

Figure 2.7: Basic neuroanatomy of the basal ganglia. SPs=spiny neurons from the striosomal compartment, SPm=spiny neurons from the matrix compartment, ST=subthalamic nuclei, DA=dopamine neurons, PD=pallidum, T=thalamus, c=cortical columns in the sensory cortex, f=cortical columns in the frontal cortex.

synaptic changes, hence allowing these structures to learn predictions and expectations about the tasks performed. This interpretation is particularly attractive since the neuroanatomy of the basal ganglia is compatible with an actor/critic architecture (Houk, Adams, & Barto, 1995; Schultz, 1998).

Actor/critic architectures and the neuroanatomy of the basal ganglia

Figure 2.7 shows the neuroanatomy of the basal ganglia (from Houk et al., 1995). As it is shown in this figure, the striatum is composed of two parts (Houk et al., 1995): circumscribed regions called *striosomes* surrounded by

tioning such as the TD model have to assume that the learning signal cannot be negative in order to account for the non-extinction of conditioned inhibitors when they are presented alone.

matrix regions. Both contain *spiny neurons* receiving inputs from the cerebral cortex and the thalamus. Spiny neurons from the striosomal regions (SPs) and from the matrix regions (SPm) differ, among other things, by the area to which they project: SPs send direct inhibitory connections to the dopamine centers from the VTA and SN and indirect excitatory connections to these same areas through the subthalamic nuclei while SPm project, with the same direct inhibitory/indirect excitatory pattern, to the pallidum which is the output of the basal ganglia (Houk et al., 1995). The pallidum could influence behavior through its projections unto the frontal cortex. Finally, dopamine neurons from the VTA and SN send projections back to the SPs and SPm.

Houk et al. (1995) argued that this anatomy is compatible with an actor/critic architecture: the *striosomal module* (cortico-striatal pathways which project unto the VTA and SN) would be the critic while the *matrix module* (cortico-striato-pallido-cortical pathways) would be the actor (see Figure 2.7). The striosomal module would use its cortical inputs to compute a prediction of reward that would be sent to the midbrain dopamine neurons. The dopamine neurons would combine them with information about the amount of primary reward collected (through yet unkown pathways probably coming from the lateral hypothalamus, see Houk et al., 1995) to generate the TD learning signal[10]. This signal would be broadcast back to the striosomal module (hence allowing it to improve its reward predictions) and to the matrix module (so its output could be adaptively driven by its cortical input). See also Houk et al., 1995 for a plausible biochemical model of synaptic plasticity for striatal neurons which would implement eligibility traces. A problem with this model is that the output of the striosomal module is not a single striatal neuron projecting unto a single dopamine neuron. Wide populations of striatal neurons are sending connections to wide populations of dopamine neurons (Houk et al., 1995). On the other hand, the dopamine responses recorded by Schultz et coll. were massive and homogenous.

Suri and Schultz (1999) used this model in the design of a real-time neural network simulating data from Schultz et coll. Its architecture is displayed in Figure 2.8. It basically combines an actor/critic architecture with some well-known techniques in neural network modeling: the actor and the critic are linear perceptrons; the behavior corresponding to the output unit with the higher activation level is emitted (noise is added to the activation of the output units so that the agent's behavior keeps some variability); the TD

[10]The direct connections would send information about $v^{\pi}[s(t)]$ since they are inhibitory while the indirect connections would send infomation about $v^{\pi}[s(t+1)]$. Experimental results show that the information from the excitatory indirect pathways reach the midbrain dopamine structures before the direct inhibitory ones. See Houk et al. (1995)

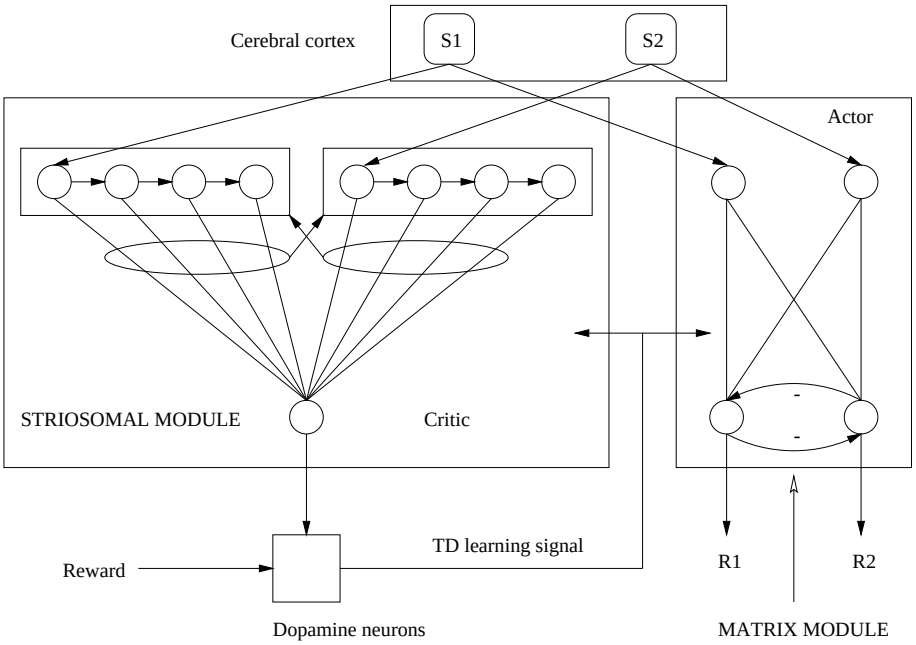

Figure 2.8: Overall organization of Suri and Schultz (1999)'s model.

learning signal computed by the learning system is used by the critic (using $TD(0)$) and the actor (using a delta rule) to update their weights.

The same model could actually have been designed without any knowledge about the underlying neurophysiological substrate if not for a few original features. So, for instance, only the critic uses tapped delay lines while a more primitive technique for the representation of temporal events is used in the actor e.g. replacing eligibility traces (see equation 1.16). This is because the time of delivery of reward does not seem to be coded at the level of the ventral striatum: if the time of delivery of an expected reward is delayed, reward expectation activity keeps on until reward delivery[11]. In the same way, the initial value of the weight between the first lag unit in a tapped delay line and the output of the critic is positive and has a lower learning rate in order to simulate the dopamine response to novel stimuli and its rather slow extinction [12]. Finally, as it can be shown on Figure 2.6, there is no depression

[11]Both input and output units have eligibility traces which are used in place of their activation level in the delta rule. Actually, the eligibility trace of an input neuron is its activation level.

[12]Because of that lower learning rate, the onset of a conditioned reinforcer in the middle of an extinction procedure initially produces a positive reward prediction followed by a negative one. This was also observed in monkeys by Schultz et coll. (see Suri & Schultz,

in dopamine activity at the usual time of delivery of a reward if this time is advanced. For Suri and Schultz (1999), this is because reinforcers and stimuli highly correlated with reinforcers "grasp the attention" of the monkey: in a sense, this caused it to forget the discriminative stimulus. In the network, if the contribution of the first lag unit from a delay line to the output of the critic is higher than that output during the last time step, the activation of all the other critic's input neurons is set to 0.

The model masters the simple tasks used by Romo and Schultz (1990) and Ljungberg et al. (1992) as well as the more complex spatial discrimination tasks used by Schultz et al. (1993). Not surprisingly, the TD learning signal mimics the dopamine neuron activity at each stage of learning in all these tasks. Interestingly, a model using the immediate amount of reward collected instead of the TD learning signal was unable not only to mimic the dopamine neuron activity but also to master the various tasks studied by Schultz et al. (1993).

Related models

Other neural models trying to take account of neurobiological constraints and using NDP to modelize the learning signal coming out from the dopamine centers are Donahoe et al. (1993)'s and Friston, Tononi, Reeke, Sporns, and Edelman (1994)'s ones. These two models actually share a lot in common like a use of complex activation functions, an emphasis on the connections between the dopamine centers and the cortex, a *selectionist* conception of learning[13] and the use of an actor/critic architecture to which an *adaptive preprocessor* has been added e.g. raw inputs are preprocessed before reaching the actor and the critic by a neural network whose weights are set by the TD learning signal generated by the critic. Another similarity which is worth mentioning is that both models assume that information about primary values is carried to the dopamine centers by the emission of the UR and not by the onset of the US, like in most models using NDP or RW-like models.

Below, we will take a closer look at Donahoe et al. (1993)'s model. Friston et al. (1994)'s model has only been used to modelize very elementary phenomena[14] while Donahoe et al. (1993)'s model has been used to address

1999).

[13]This is a view according to which learning is a process roughly analogous to natural selection: behavioral repertoire and neural populations in an individual are selected by their consequences on the environment (Donahoe & Palmer, 1994; Edelman, 1987; Skinner, 1981)

[14]It has learned a foveation reflex e.g. to bring a spotlight on the center of the retina. This spotlight has then been used as a conditioned reinforcer in a discrimination task

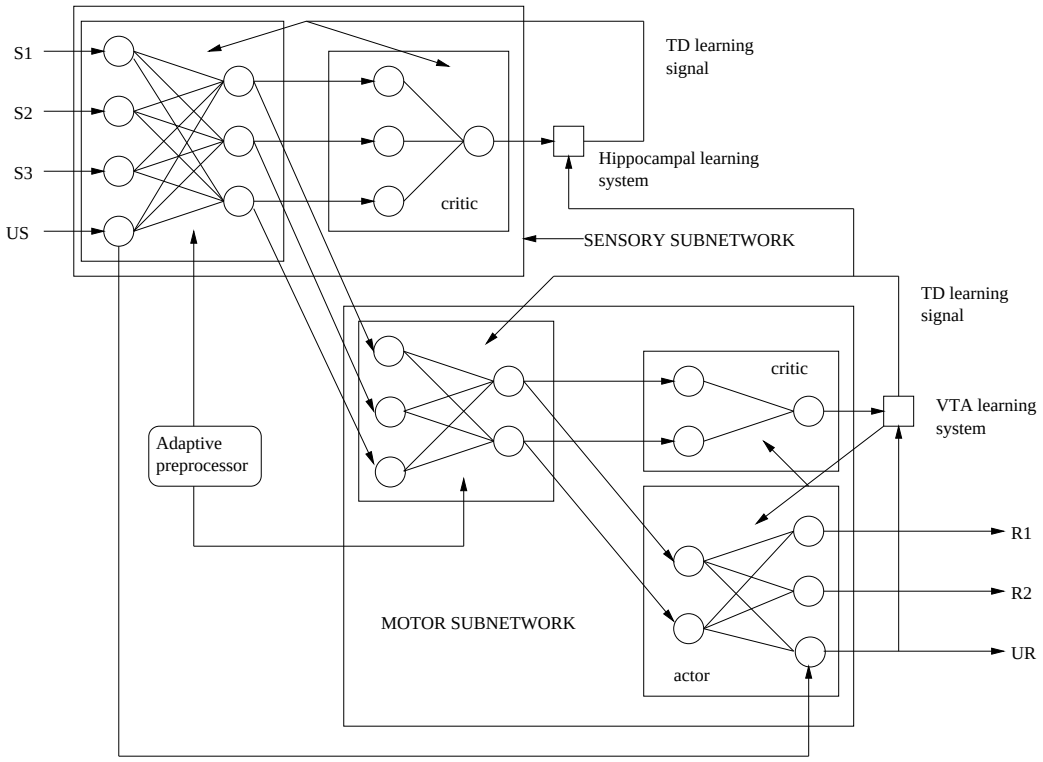

Figure 2.9: Overall organization of Donahoe et al (1993)'s model. Arrows are nonmodifiable connections: activation is simply passed from one neuron to the next one.

data about operant conditioning.

The architecture of Donahoe et al. (1993)'s model is displayed in Figure 2.9[15] but, before turning to its description, it would be useful to look at the activation functions and learning equations used by the network which are a bit more complex than the ones used in most neural network models.

- A neuron can receive signals from excitatory and inhibitory neurons. If the inhibitory signal is higher than the excitatory one, the neuron is shut down (its activation level is set to 0). Otherwise, its activation level slightly increases if the excitatory signal is above a random varying threshold. If not, it slightly decreases.

[15]This way to present Donahoe et al. (1993)'s model differs from the one used by Donahoe et al. (1993) themselves but we consider that it makes the overall functioning of the network more understandable. It is only with this kind of presentation that striking similarities between Donahoe et al. (1993)'s model and Friston et al. (1994)'s one appear.

- The learning rule takes inspiration from a neural plasticity phenomenon observed in many parts of the brain (and especially in the hippocampus) called *long term potentiation* (see Donahoe et al., 1993 for further details): the weight between two neurons is enhanced if they are coactive when a learning signal reaches the synaptic connection between the two neurons and decreases otherwise[16].

Now, we can turn to the network's architecture. It is composed of two systems, the sensory and the motor subnetworks, each designed according to an actor/critic architecture augmented by an adaptive preprocessor except that the actor is missing from the sensory subnetwork.

- The sensory subnetwork (SS) simulates activity in the sensory cortex. Raw inputs from the environment are processed by a one-layered perceptron whose output is sent to the motor subnetwork (MS) and to the critic of the SS. The output of the critic is then used to compute the TD learning signal[17] by the hippocampal learning system. It is positive each time the output is different from one time step to the other e.g. every time there is an unexpected change in the SS input.[18] The hippocampal learning system works even in the absence of reinforcement but its learning signal is potentiated when the VTA/SN learning system is active, thanks to connections between the two learning systems.

- The motor subnetwork simulates activity in the frontal (adaptive preprocessor and critic) and motor (actor) cortex. The output of the one-layered perceptron used as a preprocessor is sent to the critic which computes a reward prediction used by the VTA/SN learning system to compute the TD learning signal. The critic and the actor are also one-layered perceptrons. Each output unit of the actor stands for a behavior, its activation level being interpreted as the probability of emission of the corresponding behavior. A very special output unit is the one standing for the CR/UR since its activation level is sent back to the VTA/SN learning system to determine the amount of primary value collected at a given time step. It can be directly activated by the input unit of the SS standing for the US.

[16]The increase is proportional to the amplitude of the learning signal and to the activation level of the two neurons. The decrease is also proportional to the neurons' activation level.

[17]For the SS, the learning signal is actually the absolute value of the TD learning signal

[18]The introduction of the SS was motivated by experiments on compound conditioning in a Pavlovian context: if a compound stimulus AB is followed by the US but this is not the case for A and B alone, an animal will quickly learn to emit a CR only after AB except if its hippocampus has been destroyed.

The model provides a unifying framework for operant and Pavlovian conditioning, accounting for basic operant and Pavlovian phenomena (acquisition, extinction, discrimination, blocking,...) but each time in a purely qualitative way. It is also able in some way to account for the so-called *biological constraints on learning*. For instance, it is almost impossible to make a pigeon learn to peck a key to avoid an electric shock while it will quickly learn to press a button with its wings to avoid that shock (Bolles, 1970). In Donahoe et al. (1993)'s network, a CR also develops during an operant conditioning procedure and reaches its asymptotic level before the operant response. So, if both are incompatible, the CR will block the acquisition of the operant. This is the problem with key pecking to escape shocks in the pigeon: electric shocks elicit wing flapping movements which compete with the key pressing response. On the other hand, if the CR and the operant response are compatible, the early emergence of the CR will facilitate the acquisition of the operant. So, shock elicited wing flapping are compatible with pressing a button with a wing and this is why a pigeon learns this response so quickly.

The model is also able to simulate some features of pigeon responding under FI. The pigeons' performance in FI is characterized by the so-called *scallop* pattern of responding: after a reinforced response, the pigeon waits before starting to peck at a high rate until the next reinforced response. The duration of the pause before the pigeon starts pecking again is proportional to the interval of the schedule (Fester & Skinner, 1957; Williams, 1988; Zeiler, 1977). Submitting their model to a simulated FI 10 seconds, Donahoe and Burgos (1999) were able to synthesize scallop responding despite the absence of any explicit timing mechanisms in their system. This can be related to a study by Burgos (1997) who used a genetic algorithm[19] to make the architecture of the basic network depicted in Figure 2.9 evolve. The networks were submitted to a Pavlovian conditioning procedure with different ISI and their fitness score was the proportion of emission of the CR on the 25 last test trials. Among other results (see Burgos, 1997 for further details), Burgos (1997) discovered that the networks from the last generations were displaying the inverted U-shaped ISI function peaking at the ISI which was trained with their ancestors, despite the existence of an explicit timing mechanism. But Donahoe and Burgos (1999)'s simulations remain purely qualitative and no other interval values were explored to see if the model could account for

[19]A genetic algorithm is a program mimicking evolution through natural selection: solutions for a given problem are generated and their fitness score (how good they are to solve the problem) evaluated. The solutions with the highest fitness score are selected and a new generation of solutions is created by mixing their "genotype" (see, Mitchell, 1997 for an introduction).

more advanced properties of pigeons' responding under interval schedules.

Recently, Donahoe and Burgos (2000) and Burgos (2000) have started to modify the basic network architecture depicted in Figure 2.9. They still use the same activation and learning rules and they stick to the two subnetwork/two learning system architecture but they have changed the architecture of the subnetworks. These modifications allowed them to account for *reinforcer reevaluation* phenomena (Donahoe & Burgos, 2000) and *superstitious behaviors* (Burgos, 2000).

- Reinforcer reevaluation refers to a modification of the rate of emission of an operant induced by an independent manipulation of the reinforcer value (Donahoe & Burgos, 2000). For instance, Colwill and Rescorla (1985) caused a decrease in the rate of lever pressing of rats by pairing the food used as a reinforcer with poison outside of the operant conditioning situation. To simulate this, Donahoe et al. (1993) had to assume recurrent connections between the VTA/SN learning system and the input units of the motor subnetwork's actor. These units are highly activated at the time of reinforcement and, so, gain an excessive control over the operant response. By manipulating the weight between them and the operant response, it is then possible to modify the probability of emission of the operant. But Donahoe and Burgos (2000) do not describe how operations like pairing food with poison could have this effect on those weights.

- Superstitious behaviors are behaviors not required for reinforcement but which develop nonetheless when a reinforcer is delivered in certain conditions (Burgos, 2000; Staddon, 1977). For instance, if food is delivered to a pigeon every x seconds, the animal will develop species-specific and stereotyped behaviors during the interreinforcement interval most likely appearing after the delivery of the reward (Skinner, 1948; Staddon & Simmelhag, 1971). A related phenomenon is the non-programmed emergence of keypecking in the autoshaping procedure. Burgos (2000) has shown that if the preprocessors and actors of Donahoe et al. (1993)'s model are no more perceptrons (e.g. the connections between the input and output layers are no more extensive), then other units than the operant response unit and the CR/UR unit are activated during an operant conditioning procedure. This is a first step since other features of superstitious behaviors remain to be simulated such as their timing properties.

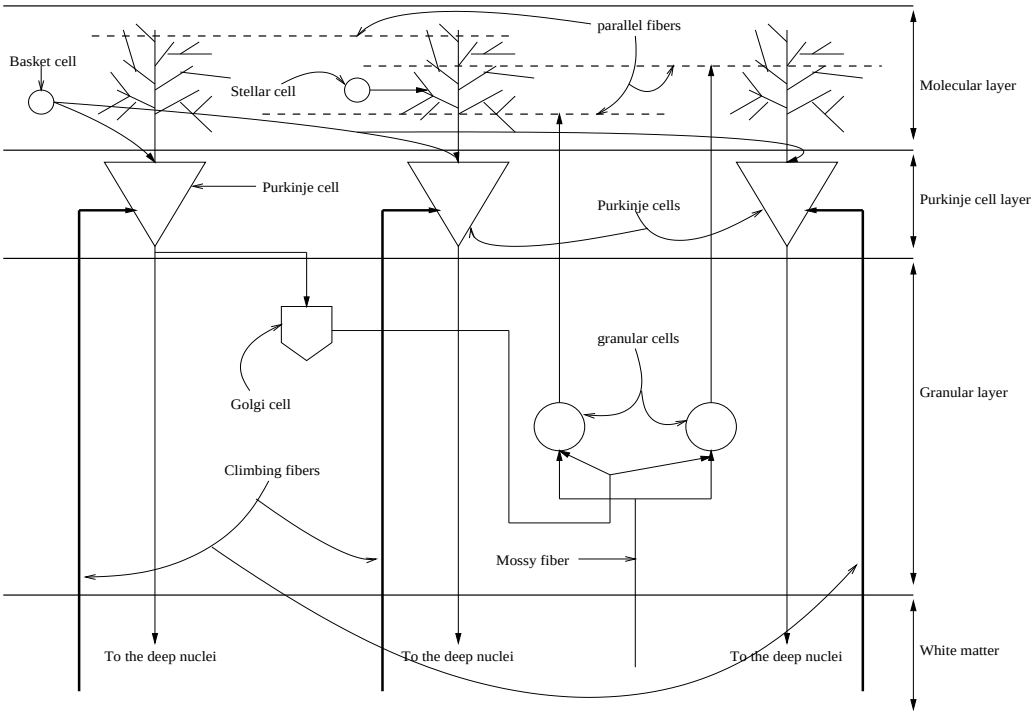

Figure 2.10: Neuroanatomy of the cerebellar cortex

2.3.2 Neurodynamic programming and the cerebellum

Neuroanatomy of the cerebellum

The dopamine systems are not the only neural structures involved in learning. Another part of the brain which has attracted a lot of attention is the cerebellum. Especially, it has been shown that the cerebellum is fundamental for NMR conditioning in the rabbit (Bartha & Thompson, 1995). Electrophysiological and lesion studies have identified the precise neural pathways necessary and sufficient for the acquisition of the CR in that context and they only imply the brain stem and the cerebellum. Higher structures play a mediating role but are not necessary, at least in the basic procedure. For instance, a rabbit without a hippocampus can acquire a CR except in a trace conditioning procedure where there is a delay between the offset of the CS and the onset of the US (Bartha & Thompson, 1995). In this part, we will review the main anatomical and neurophysiological features of the cerebellum before turning to its relations with NDP.

Just like the brain is covered by a layer of grey matter (the cerebral cortex), the cerebellum is also covered by such a layer: the *cerebellar cortex.*

Its structure is highly regular and is displayed in Figure 2.10 (Ghez, 1991). It is composed of three layers:

- The *molecular layer* mainly composed of the axons of the *granular cells* (GraCs) called the *parallel fibers* (PFs).
- The *Purkinje cell layer* contains the huge *Purkinje cells* (PCs) whose complex dendrites make contact with the PFs in the upper layer.
- The *granular layer* is composed of numerous GraCs as well as of bigger neurons called the *Golgi cells* (GCs).

Deep under the cerebellar cortex lie the deep nuclei. One of them, the *interpositus nucleus* (IP) will be central in our discussions below.

There are only two inputs to the cerebellum, both coming from various areas of the brain stem and spinal cord. They end in the cerebellar cortex but send collaterals to the deep nuclei (Ghez, 1991).

- First, we have the *climbing fibers* (CFs) which directly make contact with the PCs. The synaptic contact between the CFs and the PCs is among the strongest ones that can be found in the nervous system.
- The other input to the cerebellum and cerebellar cortex is the *mossy fibers* (MFs) which make contact with the granular cells.

The deep nuclei are the sources of all the output of the cerebellum. They project to the brain stem and spinal cord. This activity from the deep nuclei is modulated by the inhibitory influence of the PCs whose axons are the sole output of the cerebellar cortex.

Note that there are also several interconnections within the cerebellar cortex. For instance, the GCs send inhibitory projections to the GrCs (see Figure 2.10 for other examples like the inhibitory connections between the *basket cells* and the PCs). Finally, it has been observed that the contiguous activity of a PF and of the CF acting on a PC reduces the ability of this PF to trigger activity in the PC. This example of neural plasticity is called *long-term depression* (LTD).

Implementation of the TD model in the cerebellum of the rabbit

The neural pathways involved in rabbit NMR conditioning are displayed in Figure 2.11 (Bartha & Thompson, 1995; Moore & Choi, 1997; Moore et al., 1998; Rosenfield & Moore, 1995). The specific part of the cerebellar cortex involved is Larsell's HVI. Several models (see Bartha & Thompson, 1995)

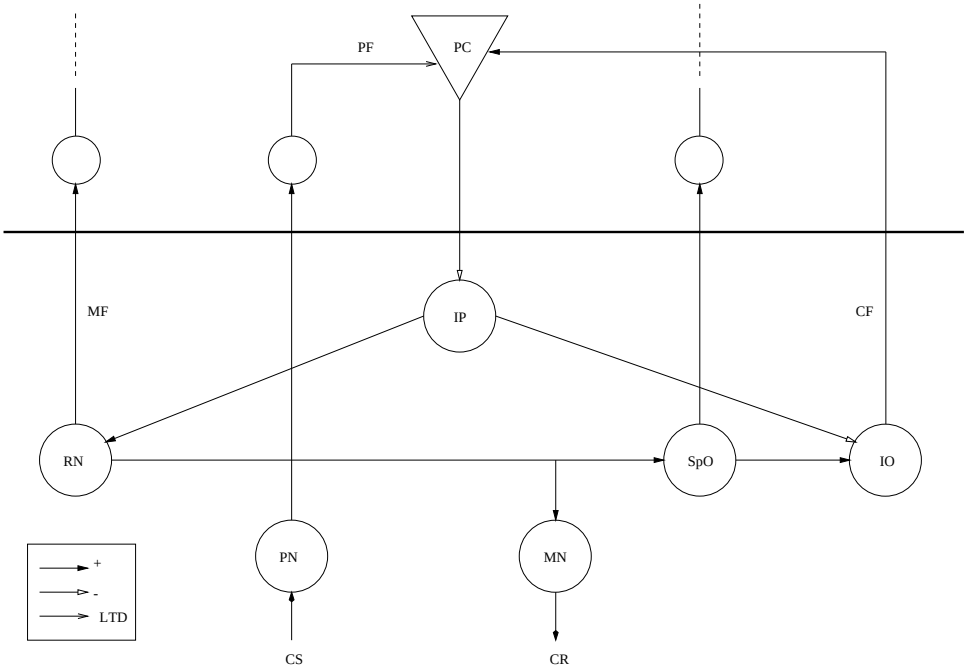

Figure 2.11: Neural pathways involved in NMR conditioning in the rabbit. (from Moore and Choi, 1997). RN=red nucleus, PN=Pontine nucleus, MN=motoneuron, SpO=spinal trigeminal nucleus, IO=inferior olive, IP=interpositus nucleus, PC=Purkinje cells, MF=mossy fibers, CF=climbing fibers. The dashed line represents the boundary of the cerebellar cortex. The empty circle above the dashed line are granular cells

have been proposed to explain the specific function played by each structure in NMR conditioning. Among these models, Moore et coll.'s one (Moore & Choi, 1997; Moore et al., 1998; Rosenfield & Moore, 1995) is particularly appealing since it is simply an implementation scheme for the TD model.

Moore et coll.'s model relies on three principles:

- A CR is triggered when the red nucleus (RN) is activated by the IP. When a CS is presented, it activates GraCs which activate PCs. This causes the inhibition of the IP activity. For a CR to be triggered, the inhibition of the IP by the PCs must be reduced through LTP. The role of the IP and of the RN in the emission of the CR has been confirmed by several studies. For instance, Desmond and Moore (1991b) and Berthier and Moore (1990) have shown that the activity in the IP and the RN is correlated with the CR although these two nuclei start firing before the emission of the CR (as it should be expected since they are

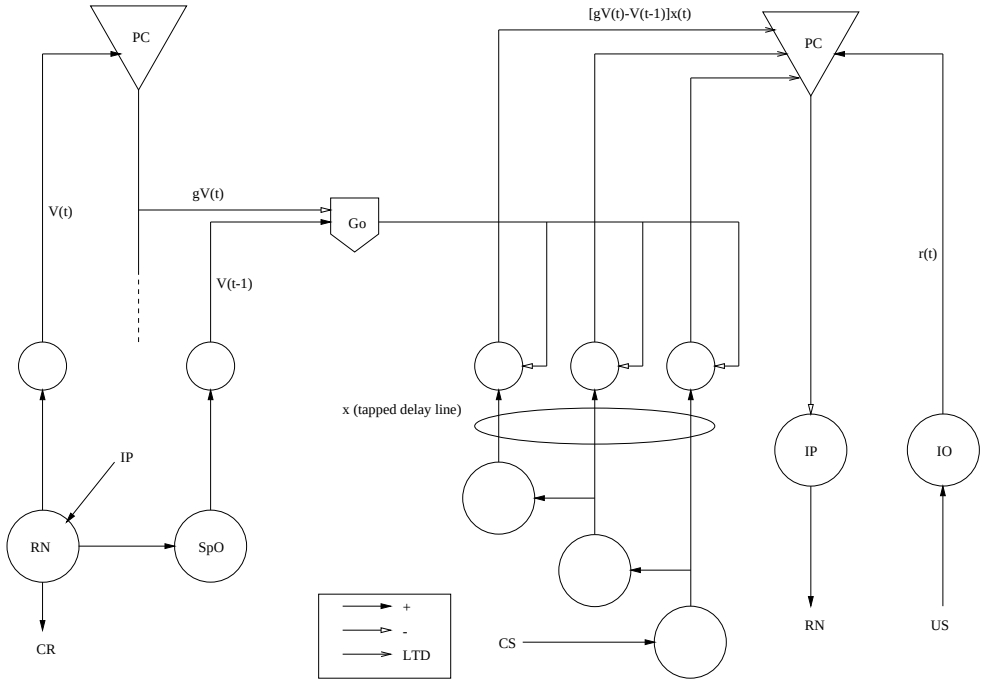

Figure 2.12: Implementation of the TD model in the cerebellum according to Moore et coll. (adapted from Moore and Choi, 1997). See the preceeding figure for the meaning of the abbreviations

premotor centers).

- LTP can also be induced by a strong activation of the PFs. This has been shown experimentally by Hartell (1996). If a NS is presented alone, it will activate the PFs but this activation is supposed to be too low to induce a LTD.

- The amplitude of the LTD depends on the amount of activity impinging on a PC. This sounds like a paradox since the consequence of LTD is to reduce the efficiency of the synaptic transmission between a PF and the PC. Like most excitatory pathways in the brain, PFs send glutamate to the PC when activated and there are actually two kinds of glutamate receptors on a PC. Experimental evidences indicate that LTP is caused by a desensitization of the response of one of these two receptors to glutamate (Crepel, Hemart, Jaillard, & Daniel, 1995). So, the amount of neurotransmitters sent to a PC when a PF is fired is not changed by LTD. The amplitude of LTD would be determined by this amount of neurotransmitter (which could be captured by the glutamate receptors

not involved in LTD).

So, the amount of activity impinging on a PC is a kind of learning signal. To show that the cerebellum could implement the TD model, it must be shown that this activity is equal to the TD learning signal $r(t) + \gamma v^\pi[s(t+1)] - v^\pi[s(t)]$. Moore et coll. consider that it is composed of two parts: the *primary reinforcement signal* $r(t)$ and the *secondary reinforcement signal* $\gamma v^\pi[s(t+1)] - v^\pi[s(t)]$. Each signal would be transmitted to the PC through different pathways (see Figure 2.12).

- The primary reinforcement signal $r(t)$ would be transmitted by the CFs activated by the inferior olive (IO). Note that the IO, which is activated by the US, is supposed to trigger the UR through different pathways from the CS. This is coherent with known neuroanatomical data.

- The real original point in Moore et coll.'s model concerns the secondary reinforcement signal. It would be implemented by a modulation by the GCs of the activity of the GraCs activated by the CS. More precisely :

  1. Since the RN controls the emission of the CR, it is plausible that its activity is equal to $v^\pi(t)$. This information about the actual situation value would be sent to some PCs (through the GraCs) as efferent copies. Hence, Berthier and Moore (1986) have observed that the activity of some PCs was positively correlated with the CR. Among these PCs, some of them started to fire before the beginning of the CR. Since an increase in the activity of a PC is incompatible with the emission of the CR, Moore et coll. assume that these PCs were actually receiving efferent copies of the CR sent by the RN. Through collaterals, these PCs would then inhibit the activity of the GCs connected to the GraCs activated by the CS. The amplitude of this inhibitory signal should be $v^\pi(t)$ but, because of all the neuronal waystations between the RN and the GCs, it would be $\gamma v^\pi(t)$.
  2. Efferent copies of the CR would also be sent by the RN to the SpO. Desmond and Moore (1991b) have observed that the SpO activity was correlated with the RN activity but with a delay of about 15 to 20 ms. So, the SpO activity would be equal to $v^\pi(t-1)$. This information would be sent (through the GraCs) to the GCs connected to the GraCs activated by the CS. Hence, Berthier and Moore (1986) observed that some of the PCs whose activity was correlated to the CR started to fire after the beginning of the CR. For Moore et coll., these PCs were activated by GraCs receiving information about $v^\pi(t-1)$ from the SpO.

3. If both the inhibitory signal coming from the RN and the excitatory signal coming from the SpO are taken into account, the activity of the GCs is modulated by a factor equal to $v^{\pi}[s(t-1)] - \gamma v^{\pi}[s(t)]$. So, the activity of the GCs is supposed to be determined by the change in the situation values. Since the TD model assumes that the situation value at time t is proportional to the amplitude of the CR at time t, a prediction that can be drawn from Moore et coll.'s model is that the activity of the GCs should be related to the change in the position of the eyelid or of the nictating membrane. This has been shown experimentally in monkeys by Van Kan, Gibson, and Houk (1993) (see also Edgley & Lidierth, 1987 with cats) but not yet in rabbits.

4. Since the connections between the GCs and the GraCs are inhibitory, the activity of the GraCs activated by the CS is modulated by a factor equal to $\gamma v^{\pi}[s(t)] - v^{\pi}[s(t-1)]$. This completes the implementation of the secondary reinforcement signal.

It is certainly possible to map any algorithm on any part of the brain and the value of an implementation scheme relies on its ability to account for data and on its heuristic value. We have already pointed to some data compatible with Moore et coll.'s model and the hypotheses sustaining the model are sufficiently precise to insure its heuristic value. Among these hypotheses, one assumes that the secondary reinforcement signal is implemented by efferent copies of the CR sent by the RN. A study by Ramnani, Hardiman, and Yeo (1995) seems to support this particular point. In this study, once the activity of the IP had been blocked by drugs, the CS was presented alone. This procedure normally causes the disappearance of the CR but, once the pharmacological blocking of the IP activity was released, the CR was still observed. According to Moore et coll.'s model, this is because the secondary reinforcement signal was not working since no information about the CR coming from the IP reached the RN. But, the model remains silent about eligibility traces.

2.4 Synthesizing complex behaviors

Skinner (see, for instance, Skinner, 1957, 1981) has argued that complex human behaviors are the emergent products of the interaction between the subject and his environment and that, if enough attention is payed to that interaction, basic learning processes such as the ones studied in conditioning experiments will prove to be sufficient to understand complex human activity

(see footnote 13 on selectionism). To conclude this review, we will look at two studies which had used artificial agents using reinforcement learning to see if, indeed, complex behaviors could emerge through the interaction of the agent with its environment.

2.4.1 Verbal learning

An enduring controversy in psychology and cognitive science is about the contribution of innate versus acquired factors in language acquisition (Pinker, 1994). Considering language as just another instance of adaptive behavior, Skinner (1957) tried to apply principles borrowed from the study of operant conditioning in animals to verbal behavior. In a violent and sometimes unfair (McCorquodale, 1970) criticism, Chomsky (1959) argued that reinforcement principles could never account for language acquisition and that it was necessary to postulate an innate species-specific device for that. This remains the dominant position in psycholinguistics (Pinker, 1994). Syntax is at the center of the arguments against a reinforcement learning approach to language acquisition. First, it seems that parental feedback ("good", "that's right",...) that could be used as reinforcers are contingent upon the truth value of the child's utterance and not upon its grammaticality (Brown & Hanlon, 1970): true but grammatically false utterances are reinforced while wrong but grammatically true utterances are not reinforced and are even sometimes punished[20] (but see Moerk, 1990 on a possible parental feedback concerning the grammaticality of the child's utterances in Brown and Hanlon's data as well as Saxton, Kulcsar, Greer, & Mandeep, 1998 for an experimental demonstration of its efficiency and Palmer, 1996 for possible subtle sources of reinforcement). Second, it is considered that the kind of rule-following behavior observed in grammar is, in principle, out of reach of a system using only reinforcement learning. A core assumption there is about the so-called "creativity" of language (Chomsky, 1959) e.g. the fact that anyone can utter or create grammatically correct sentences that have never been heard or spoken previously and so that could never have been reinforced before.

William Hutchison (see Hutchison, 1998b) has tried to simulate several verbal phenomena in an artificial neural network using RL to see if they were really out of reach of a RL system. His goal is not to modelize human data but to see if RL is sufficient to produce some basic verbal phenomena (Hutchison, 1998b).

[20] Another problem that arises is that, normally, reinforcers and punishers are identified a posteriori, based on their effect on behavior (Skinner, 1938).

Hutchison's network, called *7g*, is a linear perceptron using a Q-learning-like learning algorithm (see Hutchison, 1998a for further details).

- The network can receive input from external (light, sound,...) or internal (actual level of calories, last behavior emitted,...) sources of stimulation. Each input neuron is tied to a sensory modality and to a target stimulus value and it is maximally activated if its target stimulus is presented. A tapped delay line is associated to each input neuron.

- Each output units is associated to a behavior. The behavior corresponding to the output neuron with the highest activation level is emitted.

- The immediate amount of primary value collected is the logarithm of the actual amount of calories of the agent minus the amount of calories used to emit the behavior plus the amount of calories delivered to the agent as a reward. The logarithm function allows the simulation of deprivation and satiation: a given amount of calories is more valuable for an agent with a low level of calories than for an agent with a high level.

So, the architecture of 7g is very simple and, actually, as we have already mentioned above, there are serious limitations to what this kind of system can learn (Minsky & Papert, 1969) but Hutchison argues that these limitations can be overcome if enough attention is paid to the training procedures and their sequencing (Hutchison, 1998d). Taking inspiration from Skinner (1957), the two main algorithms he has developed and automatized (Hutchison, 1998a, 1998c) allow the training of a basic verbal repertoire:

- A fundamental notion in Skinner (1957)'s analysis is the one of a *minimal repertoire* which is a mapping between stimuli and responses. For instance, an *echoic minimal repertoire* allows the organism to repeat what he has heard by mapping auditive stimulations unto vocal responses. Other minimal repertoires allow the imitation of a visual model (by mapping visual stimulations unto motor responses) or the execution of instructions (by mapping auditive stimulations unto motor responses). Hutchison has developed an algorithm allowing the automatized training of minimal repertoires. The algorithm introduces the various stimuli of the minimal repertoire one by one, reinforcing good responses and randomizing the presentations of the stimuli in order to enhance their control over the responses (Hutchison, 1998a). The algorithm takes into account the spontaneous behavior of the agent in order to decide which stimuli to present, as it seems to be the case in

parent/child interactions (Hutchison, 1998b). This produces a faster learning than when the order of presentation of the stimuli was determined by the algorithm alone.

- According to Skinner (1957), the training of a minimal repertoire is important especially because it then allows the use of much more powerful teaching methods like the *prompt and fade* technique which is used very efficiently in applied setting by psychologists (Hutchison, 1998b) and, certainly, by all the parents. Suppose, for instance, you want to teach a child that the word for a dog is "dog"[21] . You could show him the picture of a dog, wait for the child to say "dog" spontaneously and then reinforce him. You could wait for a long time since the spontaneous occurrence of "dog" in a child's behavior is probably low. You could proceed in a much more efficient way if the child had an echoic minimal repertoire allowing him to repeat what he hears. This time, when you would present the picture of the dog, you could say "dog". Because of his minimal repertoire, the child would repeat "dog" and would be reinforced. The child would then learn to say "dog" in presence of the picture of the dog if this procedure were repeated, the intensity of the auditive prompt "dog" being reduced each time until it would be presented no more. Hutchison has been able to mimic the prompt and fade procedure in his network. His algorithm produces very fast learning without error.

Using these two algorithms, Hutchison (1996) has trained an echoic repertoire and a tact repertoire allowing the network to name 3 shapes and 3 colors. The tacts for the colors and the shapes were trained separately. Then, colored shapes were presented to the network and it was only reinforced if it named the color before the shape. This simple grammatical rule was trained with 8 colored shapes. Once this was acquired, the ninth colored shape was presented to see if the network generalized the grammatical rule. It did and named the color before the shape (see Hutchison, 1996 for further details about the learning procedure). Using Hutchison's network and his algorithms, we have reproduced this simulation with a greater number of shapes and colors. After having learnt to name 6 shapes and 6 colors, the network was explicitly taught the grammatical rule with 24 of the colored shapes. It then correctly generalized this rule when tested with the 12 remaining colored shapes. But, the conditions under which this grammatical generalization occurs are not clear yet.

[21] In Skinner (1957)'s terminology, this is called a tact. A tact is a verbal response under the control of a nonverbal stimulus. To emit a tact is said to tact the controlling stimulus.

Using the same network as the one used in the syntax learning simulation, Hutchison (1996) tried to teach it a verbal instruction of the "if condition x then action x" kind. Using the prompt and fade algorithm, he taught it to say "action 1" when hearing the word "stimulus 1" and "action 2" when hearing the word "stimulus 2". "stimulus 1" and "stimulus 2" were tacts controlled by colored shapes while "action 1" and "action 2" controlled the emission of two different behaviors. Once this was done, the network emitted the correct action in presence of both stimuli, even if these actions have never been previously reinforced in presence of these stimuli (see Hutchison, 1998b, 1998d for other simulations with 7g).

This does not demonstrate the validity of a RL approach to language nor does it show that language acquisition in humans is caused by reinforcement processes. But these simulations clearly show that one should be very careful before saying that a RL system is unable to learn this or that kind of behavior. Recent works with 7g have taken a more applied direction (Hutchison, personal communication). The output system of the network is now a realistic artificial jaw and the network is now able to receive real speech as input and Hutchison is currently trying to make it recognize words spoken by children so that it could supervise word learning with language-impaired children.

2.4.2 Reaching development

Berthier (1996) has used Q-learning to simulate the development of infant reaching. The set of states S of his model is composed of coordinates (x, y) on a plane and the agent interacts with its environment until it reaches a set of target states corresponding, for instance, to an object that must be grasped. Each time step, the agent emits a control $u_i(t) = \{dx_i, dy_i, v_i\}$ where dx_i and dv_i are the distances in the x and y directions and v_i is the movement speed. The state of the environment is then updated using the following equation

$$\begin{aligned} x(t+1) &= x(t) + dx_i + \varepsilon \\ y(t+1) &= y(t) + dy_i + \varepsilon \end{aligned} \tag{2.5}$$

where ε is a random variable following a gaussian distribution with mean 0 and standard deviation $\sqrt{kv_i^2 + 0.2}$. k is a free parameter controlling the stochasticity of the state transition. By decreasing it, Berthier (1996) is able to simulate the increased control an infant has over his arms as his motor system develops.

So, at time t, the environment is in state $s(t) = [x(t), y(t)]$ and the agent emits control $u_i = \{dx_i, dy_i, v_i\}$. The environment then goes into state $s(t+1)$

and $q^*[s(t), u_i]$, the estimation of $Q^*[s(t), u_i]$ stored by the agent is updated according to

$$q^*[s(t), u_i] := q^*[s(t), u_i] + \alpha\{c[s(t), u_i, s(t+1)] + \gamma \min_u q^*[s(t+1), u] - q^*[s(t), u_i]\} \tag{2.6}$$

which is just the standard Q-learning equation. $c(s, u, s')$ is simply the time required to move the arm from state s to state s'.

Berthier (1996) studied the evolution of the policy computed by equation (2.6) while k was progressively decreased. He found that, for a high value of k, the agent reached the target using a sequence of submovements with a high speed, while, when k was low, the agent reached the target with a single quick submovement. This corresponds to the evolution of reaching observed in infants (Berthier, 1996).

2.5 General conclusions

MDP and RL algorithms are extremely popular in artificial intelligence and we think that we have showed in this review that they have an enormous potential for understanding animal learning. But, it seems to us that this potential is underpexploited especially in the field of operant conditioning where the few proposed models have failed to address central issues of operant researches such as schedule performance. For us, it comes from the fact that the MDP framework has been totally neglected by these applications of reinforcement learning to conditioning.

The MDP framework is a kind of recipe to build a RL model. It makes explicit the decisions a modeler makes in building a model: (a) What is S? (b) What is U? (c) What is $f(s, s', u)$? (d) What kind of NDP algorithm the agent is using? (e) How does the agent solve the exploration/exploitation dilemma? (f) What kind of function approximation architecture is used by the agent? If it is a neural network, then several other questions need to be solved (g) What is the network's architecture? (h) What are the rules governing the activations of the neurons? (i) What kind of preprocessing should be used in the model? Each of these decisions will influence the model's performance in a dramatic way and it could be very hard to know exactly which of them are responsible for a given feature of the model's behavior. Moreover, it would be idealistic to think that making these decisions at random would lead to interesting models: the state space of the models is simply too large. Maybe a model built that way would be able to account for very basic phenomena but not for more sophisticated features of behavior. So, each decision must be constrained.

A solution used, for instance, by Donahoe et al. (1993) is to constrain the model's architecture with neurophysiological data but this raises several problems. First, neurally inspired models are always extremely simplified when compared to real brains (Marr, 2000) and some features of the nervous system are always left behind. Maybe some of these features are fundamental for the understanding of behavior (Hutchison, 1998d)[22]. Second, it is not clear if neurophysiological data provide enough constraints for a model. Basing their work on the same knowledge about the neural pathways underlying reinforcement, Donahoe et al. (1993) and Suri and Schultz (1999) come with a very different model's architecture. Third, neural data are not sufficient to constrain all the features of a model and some decisions still need to be made without their help. This is, for instance, the case of Donahoe et al. (1993)'s rules for the activation of neurons. Claiming the same neural plausibility for their model, Friston et al. (1994) come with very different equations.

Moreover, the emphasis on neural data has led this authors to focus on the architecture of their model and to neglect the first three questions about the nature of S, U and the return function. These questions are more important than the one concerning the architecture because they are more basic: they must be answered first. Actually, if S and U are small enough so that the curse of dimensionality can be avoided, a DP model, where only these three fundamental decisions are to be made, could be explored. This kind of model would focus the modeler's attention on questions relating the nature of the state space, action space and return function while postponing the more complex problems related the model's architecture. Actually, it could be argued that it will be easier to solve this problem once the nature of S, U and $f(s', s, u$ has been better understood. The drawback is that it leds to models of asymptotic performance but we see them as a necessary first step toward complete models which would also account for the learning of the final stable performance. At least, these models are in principle able to account for dynamic properties of asymtotic behavior such as FI scallop.

An example of the approach we propose here can actually be found in behavioral ecology where this kind of MDP modeling is used in so-called *dynamic state variable model* (Clark & Mangel, 2000; Mangel & Clark, 1988). As an illustration, let's take example of an animal whose internal state can be described by its level of energetic reserve $x \in X$ where $X = [x_{min}, x_{max}]$. X is the set of states of the environment for the MDP faced by the animal[23].

[22]Some have even argued that the features of the nervous system which are really important for an understanding of behavior are still unknown (see, for instance, Penrose, 1994).

[23]Note that X is continuous while the set of states of the environment in a MDP is discrete. But, X can be discretized. If it contains enough elements, then the optimal

Let $x(t)$ be the level of energetic reserve at decision stage t. At each decision stage t, the animal must choose to forage in patch 1 or in patch 2. Let control a be the decision to forage in patch 1 and control b the decision to forage in patch 2. So, the set of controls is $U = \{a, b\}$. Foraging in patch 1 costs y_1 amount of energy and returns y_2 amount of energy with a probability of p_1 while foraging in patch 2 costs y_4 amount of energy while allowing the collect of y_5 amount of energy with a probability of p_2. Finally, let's assume that the horizon of the MDP is finished so that the animal has only to choose in which patch to forage during T decision stages. So, Bellman's optimality equation for each control is

$$\begin{aligned} Q_t^*(x,a) &= (1-p_1)V_{t+1}^*(x-y_1) + p_1 V_{t+1}^*(x-y_1+y_2) \\ Q_t^*(x,b) &= (1-p_2)V_{t+1}^*(x-y_3) + p_2 V_{t+1}^*(x-y_3+y_4) \\ V_t^*(x) &= \max\left[Q_t^*(x,a), Q_t^*(x,b)\right] \\ V_T^*(x) &= \phi(x) \end{aligned} \tag{2.7}$$

where ϕ is a given function determining the fitness of the organism (it is called the *fitness function).* Using DP to solve the above system of equations, it is possible to know the optimal foraging strategy for a given amount of energetic reserve and at a given decision stage (see Jozefowiez, Darcheville, & Preux, 2002 for a first attempt to apply this approach to schedule performance).

strategy derived from this discretized set would be similar to the one derived from the continuous set.

References

Bartha, G. T., & Thompson, R. F. (1995). Cerebellum and conditioning. In M. A. Arbib (Ed.), *Handbook of brain theory and neural networks* (pp. 169–172). Cambridge, MA: MIT Press.

Barto, A. G., Sutton, R. S., & Anderson, C. W. (1983). Neuronlike elements that can solve difficult learning control problems. *IEEE transactions on systems, man and cybernetics*, *13*, 835–846.

Baum, W. M. (1981). Optimization and the matching law as the basis for instrumental behavior. *Journal of the experimental analysis of behavior*, *36*, 387–403.

Berthier, N. E. (1996). Learning to reach: a mathematical model. *Developmental psychology*, *32*, 804–823.

Berthier, N. E., & Moore, J. W. (1986). Cerebellar purkinje cell activity related to the classically conditioned nictating membrane response. *Experimental brain research*, *63*, 341–350.

Berthier, N. E., & Moore, J. W. (1990). Activity of cerebellar deep nuclear cells during classical conditioning of nictating membrane extension in rabbits. *Experimental brain research*, *83*, 44–54.

Bertsekas, D. P. (1995). *Dynamic programming and optimal control (2 volumes).* Belmont, MA: Athena Scientific.

Bertsekas, D. P., & Tsitsiklis, J. N. (1996). *Neuro-dynamic programming.* Belmont, MA: Athena Scientific.

Bishop, C. M. (1995). *Neural networks for pattern recognition.* Oxford: Oxford University Press.

Bolles, R. C. (1970). Species-specific defense reactions and avoidance learning. *Psychological review*, *77*, 32–46.

Brown, R., & Hanlon, C. (1970). Derivational complexity and order of acquisition in child speech. In J. R. Hayes (Ed.), *Cognition and the development of language* (pp. 11–54). New York: Wiley.

Burgos, J. E. (1997). Evolving artificial neural networks in pavlovian environments. In J. W. Donahoe & V. Packard-Dorsel (Eds.), *Neural networks model of cognition* (pp. 58–79). Amsterdam: Elsevier Science.

Burgos, J. E. (2000). Superstition in artificial networks: a case study for selectionist approach to reinforcement. *Mexican journal of behavior analysis*, *26*, 159–188.

Charnov, E. L. (1976). Optimal foraging: the marginal value theorem. *Theoretical population biology*, *9*, 129–136.

Chomsky, N. (1959). A review of "verbal behavior" by B.F. Skinner. *Language*, *35*, 26–58.

Clark, C., & Mangel, M. (2000). *Dynamic state variable models in ecology: methods and applications.* Oxford: Oxford University Press.

Colwill, R. M., & Rescorla, R. A. (1985). Postconditioning devaluation of a reinforcer affects instrumental responding. *Journal of experimental psychology: animal behavior processes*, *11*, 120–132.

Crepel, F., Hemart, N., Jaillard, F., & Daniel, H. (1995). Long-term depression in the cerebellum. In M. A. Arbib (Ed.), *Handbook of brain theory and neural networks* (pp. 560–563). Cambridge, MA: MIT Press.

Crites, R. H., & Barto, A. G. (1996). Improving elevator performance using renforcement learning. In D. S. Touretky, M. C. Mozer, & M. E. Hasselmo (Eds.), *advances in neural information processing systems: proceedings of the 1995 conference* (pp. 1017–1023). Cambridge, MA: MIT Press.

Desmond, J. E., & Moore, J. W. (1991a). Alterning the synchrony of the stimulus trace: test of a neural network model. *Biological cybernetics*, *65*, 161-169.

Desmond, J. E., & Moore, J. W. (1991b). Single-unit activity in read nucleus during the classically conditioned rabbit nictating membrane response. *Neuroscience research*, *10*, 260–279.

Dickinson, A., & Charnock, D. J. (1985). Contingency effect with maintained instrumental reinforcement. *Quarterly journal of experimental psychology*, *37B*, 397–416.

Dickinson, A., & Mulatero, C. W. (1989). Reinforcer specificity of the supression of the instrumental performance on a non-contingent schedule. *Behavior processes*, *19*, 167–180.

Donahoe, J. W., & Burgos, J. E. (1999). Timing without a timer. *Journal of the experimental analysis of behavior*, *71*, 257–263.

Donahoe, J. W., & Burgos, J. E. (2000). Behavior analysis and reevaluation. *Journal of the experimental analysis of behavior*, *74*, 331–346.

Donahoe, J. W., Burgos, J. E., & Palmer, D. C. (1993). A selectionist approach to reinforcement. *Journal of the experimental analysis of behavior*, *60*, 17–40.

Donahoe, J. W., & Palmer, D. C. (1994). *Learning and complex behavior.* Needham Height, MA: Allyn and Bacon.

Dorigo, M., & Columbetti, M. (1994). Robot shaping: developing autonomous agents through learning. *Artificial intelligence*, *70*, 321–370.

Edelman, G. M. (1987). *Neural darwinism.* New York: Basic Books.

Edgley, S. A., & Lidierth, M. (1987). Discharges of cerebellar golgi cells during locomotion in cats. *Journal of physiology (London)*, *392*, 315–332.

Fester, C. B., & Skinner, B. F. (1957). *Schedules of reinforcement.* New York: Appleton Century Croft.

Friston, K. J., Tononi, G., Reeke, N., Sporns, O., & Edelman, G. M. (1994). Value-dependent selection in the brain: simulation in a synthetic neural model. *Neuroscience*, *59*, 229–243.

Ghez, C. (1991). The cerebellum. In E. R. Kandel, J. H. Schwartz, & T. M. Jessel (Eds.), *Principles of neural sciences, 3d edition* (pp. 626–646). East Norwalk, CO: Appleton and Lange.

Hall, G. (2002). Associative structures in pavlovian and instrumental conditioning. In C. R. Gallistel & ?? (Eds.), *Stevens' handbook of experimental psychology (3rd edition), volume 2: learning and motivation* (pp. 1–46). New York: Wiley.

Hammond, L. J. (1980). The effect of contingencies upon appetitive conditioning of free operant behavior. *Journal of the experimental analysis of behavior*, *34*, 297–304.

Hammond, L. J., & Weinberg, M. (1984). Signaling unearned reinforcers removes supression produced by zero correlation in an operant paradigm. *Animal learning and behavior*, *12*, 371–374.

Hartell, N. A. (1996). Strong activation of parallel fibers produces localized calcium transients and a form of long-term depression that spreads to distant synapses. *Neuron*, *16*, 601–610.

Haykins, S. (1999). *Neural networks: a comprehensive foundation* (2nd ed.). Upper Saddle River, NJ: Prentice Hall.

Hearst, S. E. (1988). Fundamentals of learning and conditioning. In R. C. Atkinson, R. J. Herrnstein, G. Lindzey, & R. D. Luce (Eds.), *Stevens'handbook of experimental psychology (2nd edition), vol 2: learning and cognition* (pp. 3–109). New York: Wiley.

Hollerman, J. R., & Schultz, W. (1998). Dopamine neurons report an error in the temporal prediction of reward during learning. *Nature neuroscience*, *1*, 304–309.

Houk, J. C., Adams, J. L., & Barto, A. G. (1995). A model of how the basal ganglia generate and use neural signals that predict reinforcement. In J. C. Houk, J. L. Davis, & D. G. Beiser (Eds.), *Models of information processing in the basal ganglia* (pp. 249–270). Cambridge, MA: MIT Press.

Hutchison, W. R. (1996). *How language and cognition emerge in a behavior-based system.* Simulation of adaptive behavior conference, Cape Cod.

Hutchison, W. R. (1998a). [Adaptive autonomous agent with verbal learning]. US patent number 802-506.

Hutchison, W. R. (1998b). Computer simulations of verbal behavior for research and persuasion. *The analysis of verbal behavior*, *15*, 117–120.

Hutchison, W. R. (1998c). *Hierarchy of methods to train complex behaviorial repertoires.* Manuscript non-publi.

Hutchison, W. R. (1998d). We also need complete behavioral models. *Journal of the experimental analysis of behavior*, *67*, 224–228.

Jozefowiez, J., Darcheville, J. C., & Preux, P. (2002). Using Markovian decision problems to analyze animal performance in random and variable ratio schedules of reinforcement. In B. Hallam, D. Floreano, J. Hallam, G. Hayes, & J. A. Meyer (Eds.), *From animals to animats 7: proceedings of the seventh international conference on the simulation of adaptive behavior* (pp. 205–214). Cambridge, MA: MIT Press.

Kamin, L. J. (1969). Predictability, surprise, attention and conditioning. In B. A. Campbell & R. M. Church (Eds.), *Punishment and aversive behaviors* (pp. 279–296). New York: Appleton Century Croft.

Kehoe, E. J. (1990). Classical conditioning: fundamental issues for adaptive network models. In M. Gabriel & J. Moore (Eds.), *Learning and computational neuroscience* (pp. 389–420). Cambridge, MA: MIT Press.

Kehoe, E. J., Schreurs, B. G., & Graham, P. (1987). Temporal primacy overrides prior training in prior compound conditioning in the rabbit's nictating membrane response. *Animal learning and behavior*, *15*, 455–464.

Kremer, E. F. (1978). The Rescorla-Wagner model: losses in associative strengh in compound conditioning stimuli. *Journal of experimental psychology: animal behavior processes*, *4*, 22–36.

Ljungberg, T., Apicella, P., & Schultz, W. (1992). Response of monkey dopamine neurons during learning of behavioral reactions. *Journal of neurophysiology*, *67*, 145–163.

Mangel, M., & Clark, C. W. (1988). *Dynamic modeling in behavioral ecology.* Princeton, NJ: Princeton University Press.

Marr, M. J. (2000). What is the net worth? Some thoughts on neural networks and behavior. *Mexican journal of behavior analysis*, *26*, 273–287.

Matzel, L. D., Held, F. P., & Miller, R. R. (1988). Information and expression of simultaneous and backward association: implication for contiguity theory. *Learning and motivation*, *16*, 398–412.

McCorquodale, K. (1970). On Chomsky's review of Skinner's "verbal behavior". *Journal of the experimental analysis of behavior*, *13*, 83–99.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of ideas immanent in nervous activity. *Mathematical biophysics*, *5*, 115–133.

Millenson, J., Kehoe, E. J., & Gormezano, N. J. (1977). Classical conditioning of the rabbit's nictating membrane response under fixed and mixed cs-us intervals. *Learning and motivation*, *8*, 351–366.

Miller, R. R., Barnett, R. C., & Grahame, N. J. (1995). Assessment of the Rescorla-Wagner model. *Psychological bulletin*, *117*, 363-386.

Minsky, M. L., & Papert, S. A. (1969). *Perceptrons.* Cambridge, MA: MIT Press.

Mitchell, M. (1997). *An introduction to genetic algorithm.* Cambridge, MA: MIT Press.

Moerk, E. L. (1990). Three-term contingency patterns in mother-child interactions during first-language acquisition. *Journal of the experimental analysis of behavior*, *54*, 293–305.

Moore, J. W., & Choi, J. S. (1997). The TD model of classical conditioning: response topography and brain implementation. In J. W. Donahoe & V. Packard-Dorsel (Eds.), *neural networks models of cognition* (pp. 387–405). Amsterdam: Elsevier.

Moore, J. W., Choi, J. S., & Brunzell, D. H. (1998). Predictive timing under temporal uncertainty: the TD model of the conditioned response. In D. A. Rosenbaum & C. E. Collier (Eds.), *Timing of behavior: neural, computational and psychological perspective* (pp. 3–34). Cambridge, MA: MIT Press.

Palmer, D. C. (1996). Achieving parity: the role of automatic reinforcement. *Journal of the experimental analysis of behavior*, *65*, 289–290.

Parker, G. A., & Maynard Smith, J. (1990). Optimality theory in evolutionary biology. *Nature*, *348*, 27–33.

Pavlov, I. P. (1927). *Conditioned reflexes.* New York: Oxford University Press.

Penrose, R. (1994). *Shadows of the mind: a search for the missing science of consciousness.* Oxford: Oxford University Press.

Pinker, S. (1994). *The language instinct.* New York: Harper Collins.

Ramnani, N., Hardiman, M. J., & Yeo, C. H. (1995). Temporary inactivation of the cerebellum prevents the extinction of conditioned nictating membrane responses. *Society for neuroscience abstracts*, *21*, 1222.

Rescorla, R. A. (1968). Probability of shock in the presence and absence of CS in fear conditioning. *Journal of comparative and physiological psychology*, *66*, 1–5.

Rescorla, R. A. (1971). Variation in the effectiveness of reinforcement and nonreinforcement following prior inhibitory conditioning. *Learning and motivation*, *2*, 113–123.

Rescorla, R. A., & Wagner, A. R. (1972). A theory of Pavlovian conditioning: variations in the effectiveness of reinforcement and nonreinforcement. In A. H. Black & W. F. Prokazy (Eds.), *classical conditioning II* (pp. 64–99). New York: Appleton Century Croft.

Robbins, T. W., & Everitt, B. J. (1996). Neurobehavioral mechanisms of reward and motivation. *Current opinion in neurobiology*, *6*, 228–236.

Romo, R., & Schultz, W. (1990). Dopamine neurons of the monkey midbrain: contingencies of responses to active touch during self-initiated arm movements. *Journal of neurophysiology*, *63*, 228–236.

Rosenfield, M. A., & Moore, J. W. (1995). Connections to cerebellar cortex (Larsell's HVI) in the rabbit: a WGA-HRP study with implications for classical eyeblink conditioning. *Behavioral neuroscience*, *19*, 1106–1118.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart & J. L. McClleland (Eds.), *Parallel distributed processing, exploration into the microstructure of cognition, vol 1: foundations* (pp. 318–362). Cambridge, MA: MIT Press.

Saxton, M., Kulcsar, B., Greer, M., & Mandeep, R. (1998). Long-term effect of corrective input: an experimental approach. *Journal of child language*, *25*, 701–721.

Schanks, D. R. (1994). Human associative learning. In N. J. Mackintosh (Ed.), *Animal learning and cognition* (pp. 335–374). Cambridge, MA: Academic Press.

Schmajuk, N. A. (1997). *Animal learning and cognition: a neural network approach.* Cambridge, UK: Cambridge University Press.

Schultz, W. (1998). Predictive learning signal of dopamine neurons. *Journal of neurophysiology*, *80*, 1–27.

Schultz, W., Apicella, P., & Ljungberg, T. (1993). Response 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., Dayan, P., & Montague, R. R. (1997). A neural substrate of prediction and reward. *Science*, *275*, 1593–1599.

Schultz, W., Tremblay, L., & Hollerman, J. R. (1998). Reward prediction in primate basal ganglia and frontal cortex. *Neuropharmacology*, *37*, 421–429.

Siegel, S., & Allan, L. (1996). The widespread influence of the Rescorla-Wagner model. *Psychonomic bulletin and society*, *3*, 421–429.

Singh, S. P., & Sutton, R. S. (1996). Reinforcement learning with replacing eligibility traces. *Machine learning*, *22*, 123–158.

Skinner, B. F. (1938). *The behavior of organisms.* New York: Appleton Century Croft.

Skinner, B. F. (1948). "Superstition" in the pigeon. *Journal of experimental psychology*, *52*, 270–277.

Skinner, B. F. (1957). *Verbal behavior.* New York: Appleton Century Croft.

Skinner, B. F. (1981). Selection by consequences. *Science*, *213*, 501–504.

Staddon, J. E. R. (1977). Schedule-induced behavior. In W. K. Honig & J. E. R. Staddon (Eds.), *Handbook of operant behavior* (pp. 125–152). Englewood Cliffs, NJ: Prentice-Hall.

Staddon, J. E. R. (1983). *Adaptive behavior and learning.* Cambridge, MA: Cambridge University Press.

Staddon, J. E. R., & Honig, W. K. (1977). *Handbook of operant behavior.* Englewood Cliffs, NJ: Prentice-Hall.

Staddon, J. E. R., & Simmelhag, V. L. (1971). The "superstition" experiment: a reevaluation of its implications for the principles of adaptive behavior. *Psychological review*, *78*, 3–43.

Suri, R. E., & Schultz, W. (1999). A neural network model with dopamine-like reinforcement signals that learns a spatial delayed response task. *Neuroscience*, *91*, 871–890.

Sutton, R. S. (1988). Learning to predict by the method of temporal differences. *Machine learning*, *3*, 9–44.

Sutton, R. S., & Barto, A. G. (1981). Toward a modern theory of adaptive network: expectation and prediction. *Psychological review*, *88*, 135–170.

Sutton, R. S., & Barto, A. G. (1990). Time-derivative models of Pavlovian reinforcement. In M. Gabriel & J. Moore (Eds.), *learning and computational neuroscience* (pp. 497–537). Cambridge, MA: MIT Press.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: an introduction.* Cambridge, MA: MIT Press.

Tessauro, G. J. (1994). TD-gammon, a self-teaching backgammon program, achieves master-level play. *Neural computation*, *8*, 257–277.

Tremblay, L., Hollerman, J. R., & Schultz, W. (1998). Modifications of reward expectation-related neuronal activity during learning in primate striatum. *Journal of neurophysiology*, *80*, 964–977.

Van Kan, P. L. E., Gibson, A. R., & Houk, J. C. (1993). Movement-related inputs to intermediate cerebellum in the monkey. *Journal of neurophysiology*, *69*, 74–94.

Wagner, A., Logan, F., Haberlandt, K., & Price, T. (1968). Stimulus selection in animal discrimination learning. *Journal of experimental psychology*, *76*, 171-180.

Wasserman, E. A., & Miller, R. R. (1997). What's elementary about associative learning? *Annual review of psychology*, *48*, 573-607.

Watkins, C. J. C. H., & Dayan, P. (1992). Q-learning. *Machine learning*, *8*, 279–292.

Widrow, B., & Hoff, M. (1960). Adaptive switching circuit. *IRE WESCON Convention record*, 96–104.

Williams, B. A. (1975). The blocking of reinforcement control. *Journal of the experimental analysis of behavior*, *24*, 215–225.

Williams, B. A. (1988). Reinforcement, choice and response strength. In R. C. Atkinson, R. J. Herrnstein, G. Lindzey, & R. D. Luce (Eds.), *Stevens' handbook of experimental psychology, volume 2: learning and cognition* (pp. 167–244). New York: Wiley.

Wise, R. A. (1996). Addictive drugs and brain stimulation reward. *Annual review of neuroscience*, *19*, 319–340.

Wise, R. A., & Rompre, P. P. (1989). Brain dopamine and reward. *Annual review of psychology*, *40*, 191–225.

Zanich, M. L., & Fowler, H. (1978). Transfer from pavlovian conditioning to instrumental appetitive learning: signaling versus discrepancy interpretation. *Journal of experimental psychology: animal behavior processes*, *4*, 37–49.

Zeiler, M. D. (1977). Schedules of reinforcement: a theoretical analysis. In W. K. Honig & J. E. R. Staddon (Eds.), *Handbook of operant behavior* (pp. 201–233). Englewood Cliffs, NJ: Prentice-Hall.