This chapter appears in
The Computing Neuron, R. Durbin,
C. Miall and G. Mitchison (Eds.),
Addison-Wesley: Wokingham, England, 1989.

5
From Chemotaxis to Cooperativity: Abstract Exercises in Neuronal Learning Strategies

Andrew Barto

Summary

In this chapter I draw on parallels that have been made between neurons and free-living unicellular organisms, to explore the idea that basic neural learning mechanisms have an abstract structure similar to that of the chemotactic behavior of certain free-living unicellular animals. The major line of support that I bring to bear on this hypothesis consists of theoretical and computational results relating to the collective behavior of neuron-like units that implement the suggested learning mechanism. These results show that such neuron-like units are capable of learning how to behave as effective decision makers in distributed systems and are able to learn how to cooperate with one another to solve problems that the individual units are not able to solve. The considerable range of collective behavior that emerges from a few simple principles, suitably refined, suggests that behavior patterns seen in unicellular organisms might serve as models for aspects of the learning capabilities of mature neurons.

5.1 Introduction

A large part of contemporary computer science is devoted to the study of parallel and distributed processing. Although parallel processing and distributed processing are often not distinguished, computer scientists do make a distinction between them that, while not being completely sharp, nicely encompasses the perspective I take in this chapter. When multiple processors cooperate closely to perform a task that somehow has been divided among them, the term parallel processing applies. The term distributed processing, on the other hand, applies when multiple processors cooperate more loosely in performing separate tasks (Kleinrock, 1985). In the first case, a problem's potential for concurrency is exploited to achieve a faster solution; whereas in the second, distribution is forced on the system by natural circumstances. Distributed processing systems come about, for example, when the data do not arise from a localized source and

decisions, which may not act through a central authority, have to depend on remotely generated data and have outcomes dependent on decisions made at other sites.

Neural networks and neurally-inspired connectionist systems are generally seen as exemplifying principles of parallel processing. Indeed, these are examples *par excellence* of systems using closely coupled processors to carry out complex computations with remarkable speed. I can hardly disagree with this categorization, but having pointed out that the distinction described above is not sharp, in this chapter I emphasize aspects of neural networks that are more closely akin to those emphasized in distributed processing. In order to do this, it is necessary to shift from viewing a neural network as a massively parallel computational system to viewing a network as a confederation of units that face special difficulties in achieving their local goals because of their spatial distribution, limited means of communication, and lack of access to centralized control information. This shift in perspective requires some idea of what a unit's local task is so that it makes sense to think of a network as a distributed processing system. We have to modify the assumptions often made in studies of theoretical neural networks, that the processing units are *simple* and that nearly everything of interest in a network is the result of the interaction of many units.

In this chapter I draw on parallels that have been made between neurons and unicellular organisms to explore the idea that basic neural learning mechanisms have an abstract structure similar to that of the adaptive strategies possessed by certain free-living unicellular animals. One of the motivations for studying unicellular organisms, such as *Paramecium* and certain kinds of bacteria, is that, while being relatively easy to study, there are similarities between the cellular mechanisms of these organisms and those of excitable cells such as neurons (Koshland, 1980; Hinrichsen & Schultz, 1988). One factor making these organisms useful models is that because they are free-living, correspondences between cellular mechanisms and cellular behavior can be particularly clear, as when, for example, depolarization causes *Paramecium* to swim backwards (Hinrichsen & Schultz, 1988). Most important for my purposes, however, is the apparent clarity with which one can attribute functional significance to cellular behavior and thereby establish correspondences, albeit speculative ones, between mechanisms and the adaptive utility of behavior. The backward swimming of *Paramecium*, for example, plays a role in an avoidance response, and the swimming of the bacterium *Escherichia coli* can be understood in terms of chemotactic behavior that causes the bacterium to approach and remain in the vicinity of nutrients (Koshland, 1980).

Not only do behavioral patterns such as these remind us that single cells can exhibit subtle and complex behavior, these patterns can be abstracted and elaborated to serve as the basis for a neuronal model of learning. According to this model, instead of modulating swimming behavior, neurons adjust their firing tendencies under the influence of changing levels of substances analogous to the attractants and repellents that modulate bacterial movement. Hence, the suggestion is that chemotactic-like mechanisms operate in neurons, not just during development

where neurotrophic factors synthesized in target fields influence neuron growth and survival, but in mature neurons where they manifest themselves not as movement strategies, but as mechanisms underlying learning.

The major line of support that I can bring to bear on this hypothesis consists of theoretical and computational results relating to the *collective* behavior of neuron-like units that implement the proposed learning mechanism, and much of what follows is devoted to describing these results and their place within the relevant theoretical traditions. These results show that neuron-like units, implementing learning rules of the type suggested by chemotactic behavior, are capable of learning how to behave as effective decision makers in distributed systems. They are able to make progress toward achieving local goals in spite of the uncertainty produced by their limited local views and their limited ability to control important system variables. Most important is the fact that distributed systems composed of this type of unit are able to learn how to solve problems that the individual units are not able to solve. A form of cooperativity emerges from the collective behavior of these units that may be of significance in helping to understand aspects of the adaptive behavior of aggregates of cells.

Despite my emphasis on the collective behavior of relatively complex neuron-like units with capabilities motivated by the study of unicellular organisms, it is not my intention to argue that nervous systems are colonies of loosely interacting, autonomous cells, or that neurons and chemotactic unicellular organisms are particularly closely related in an evolutionary sense. Aspects of the behavior of neural networks might be more understandable if our view shifted a bit toward the view that individual neurons have their own agendas, but no adequate account of real neural networks can ignore the specialized nature of neurons or the high degree of network structure present in nervous systems.

The perspective presented here has developed over a number of years, beginning with the hypothesis of Klopf (1972, 1982) that many aspects of memory, learning, and intelligence can be understood by viewing neurons as self-interested agents, and continuing with my research in collaboration with a number of colleagues (e.g. Barto, 1985; Barto *et al.*, 1986; Barto & Anderson, 1985; Barto, 1986; Barto & Jordan, 1987). I also draw heavily on the research of M.L. Tsetlin, the Soviet cybernetician whose studies of learning automata and their collective behavior conducted in the 1960s appeared in English in 1973 (Tsetlin, 1973). Some of the examples I use were presented by Tsetlin in a 1965 address to a section of the Physiological Society (appears in English in Tsetlin, 1973). I use these examples to draw attention to Tsetlin's work, which is not widely enough known, and because I know of no better examples. This line of research has continued as the theory of stochastic learning automata, reviewed by Narendra & Thathachar (1974). I also draw heavily of the research of D.E. Koshland on bacterial chemotaxis and its relation to neurobiology (Koshland, 1980) and of Yu-Chi Ho (1980) on decentralized decision making. I omit technical details as much as possible in this presentation, leaving the interested reader to consult the references.

5.2 Neurons and unicellular organisms

Providing biological motivation for the learning methods to be considered are the capacities of free-living unicellular organisms for goal-directed motility and similarities that have been noted between these organisms and neurons. Much of the study of single-cell movement is based on its potential relevance to cell motility in multicellular organisms, for example, to the migration of embryonic cells in morphogenesis and epithelial cells in wound healing. Other studies, however, are motivated by the relevance of unicellular organisms as models of the membrane mechanisms and behavior of excitable cells. For example, one reason the ciliated protozoan *Paramecium* is being studied is because it responds to sensory stimulation by Ca^{2+}-mediated depolarization which causes backward swimming by reversing the ciliary beat. With repolarization, the cells swim forward again but in a new direction. Overall, the behavioral response seems to be regulated by eight different ionic conductances together with the participation of cAMP and cGMP (Hinrichsen & Schultz, 1988).

Bacteria such as *Escherichia coli*, *Bacillus subtilis*, and *Salmonella typhimurium* are also studied for their potential relevance to neurobiology. The work of Koshland (Koshland,1980) most clearly explores the possible relationship between bacterial chemotaxis and neuronal function. Even though they are not eukaryotes, bacteria share many of the properties of neurons. Like neurons, they have chemical receptors for sensing their environments, they contain a signal processing system of moderate complexity, and they produce a response: release of neurotransmitter in the neuron's case and the reversal of flagellar rotation in that of the bacteria. Bacteria also show adaptation, memory, and sensory integration that may resemble those processes in neurons (Koshland,1980). Lackie (1986) provides additional information on bacterial chemotaxis and other movement strategies of single cells.

Most relevant to the neuronal learning method I describe below is the chemotactic behavior of bacteria such as those studied by Koshland. Such a bacterium propels itself along a relatively straight path by rotating its flagella, which form a bundle. Upon reversing the direction of flagellar rotation, the bundle becomes disorganized, which causes the bacterium to stop and tumble in place. As the flagella continue to rotate in this new direction, they reorganize and cause the bacterium to again be propelled on a straight path but in a new, randomly determined, direction. A kind of chemotaxis, which is the directed response to a chemical substance in the environment, results because the frequency of flagellar reversal is modulated by movement with respect to levels of attractant and repellent chemicals. Reversal frequency decreases if movement is toward higher attractant concentrations and increases if movement is toward lower concentrations. Repellents have the opposite effect. This modulation of flagellar reversal biases locomotion so that the bacterium finds, and remains near, places of maximal attractant concentration or minimal repellent concentration. It is an effective strategy, particularly when the gradient information is very noisy.

This behavior pattern has been called the 'Run-and-Twiddle Strategy': if things are getting better, keep doing what you are doing; if things are getting worse, do something else (Selfridge, in press). It is called klinokinesis with adaptation when played out over space by bacteria. Klinokinesis refers to the alteration of the direction of travel, and adaptation refers to the fact that the sensory system responds to changes in concentration rather than to absolute concentrations. We may also recognize in 'Run-and-Twiddle' a variant of the 'Win-Stay / Lose-Shift' strategy that has been studied by psychologists. In this case, winning and losing respectively correspond to swimming towards higher and lower attractant concentrations (or lower and higher repellent concentrations). An important distinction between the 'Run-and-Twiddle' behavior of bacteria and the usual form of 'Win-Stay / Lose-Shift' is that the latter behavior pattern is deterministic whereas the former is probabilistic. It is the *probability* of changing swimming direction (i.e. of twiddling or shifting) that is increased by losing, and it is the *probability* of swimming in the same direction (i.e. of running or staying) that is increased by winning. As I discuss below, the probabilistic nature of this behavior pattern is important because it has advantages when uncertainty is present in the task.

Koshland (1980) proposed a model for the regulation of bacterial tumbling that is relevant here because it suggests how aspects of the neuronal learning rule I describe below could be produced by neurons. That is, one can postulate a mechanism for altering neuronal firing rate in response to afferent signals that is analogous to how a bacterium's sensory signals may alter the frequency of flagellar reversal. Although the details of how a model of tumble regulation might apply to neurons has to await the description of the neuronal learning rule given below, it is appropriate to describe Koshland's model of tumble regulation here. According to this model, flagellar reversal is controlled by the concentration of a chemical response regulator, X, that ordinarily varies randomly about a background level X_{crit}. When the concentration of X is above X_{crit}, tumbling is suppressed; when below X_{crit}, tumbling frequency is enhanced (Figure 5.1). The response regulator, X, is formed from a substrate, W, with a rate V_f, and decomposed with rate V_d to product Y. Ordinarily, V_f approximately equals V_d, so that the concentration of X shows a small variation about a fixed level. This shared value of V_f and V_d is, in turn, regulated by the rate at which attractant and repellent molecules bind to receptors. This is hypothesized to occur by means of alterations in the reactions that produce the enzymes required for the formation and decomposition of X. Swimming through higher attractant concentrations results in higher, but still equal, values for V_f and V_d. Higher repellent concentrations result in lower values for V_f and V_d. Thus, the absolute levels of attractant and repellent concentrations do not influence the background concentration of X and hence do not alter tumble frequency, as is appropriate to describe the behavior of bacteria sensing uniform levels of attractants and repellents over time.

However, in the model it is assumed that the *rates* at which V_f and V_d change in response to *changes* in attractant and repellent concentrations differ, with V_f changing more quickly than V_d . Thus, upon sensing an increase in attractant

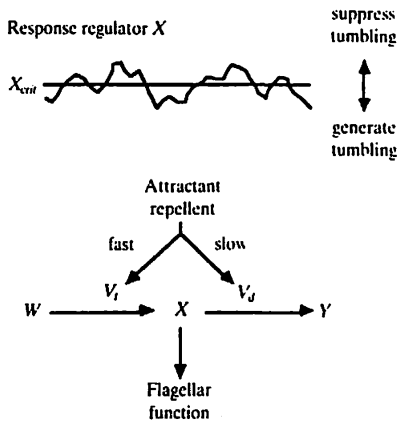

Figure 5.1 A model for bacterial chemotaxis from Koshland (1980).

concentration, the concentration of X undergoes a transient increase due to V_f being temporarily larger than V_d. This transiently suppresses tumbling. Similarly, decreases in attractant concentration, or increases in repellent concentrations, cause transient decreases in the concentration of X and increases in tumble frequency. As long as attractant concentrations are increasing, as when the bacterium is swimming up an attractant hill, tumbling remains suppressed as V_d lags behind V_f. Similarly, tumbling is enhanced upon swimming down an attractant hill. According to this model, then, both the adaptation to constant levels of attractant and repellent concentrations and the short-term memory required to provide the information needed to determine attractant and repellent gradients are produced by the dynamics of the cellular biochemistry and not (of course) by networks of cells.

5.3 Control and uncertainty

The first step in translating and extending chemotactic-like behavior patterns into specific hypothetical neuronal learning rules is to place these behavior patterns within a theoretical context that can help us extract their fundamental features. In trying to place the bacterium's klinokinesis with adaptation within a theoretical context, one first regards it as a simple hillclimbing method, as indeed it is, but to stop there obscures some of the issues involved in understanding the nature of this behavior. The hillclimbing consequences of the bacterium's behavior pattern are part of a global view that goes beyond the local task faced by the organism from moment to moment. Probabilistic 'Win-Stay / Lose-Shift' is an appropriately

microscopic characterization of the *strategy*, but what kind of *task* is faced locally by the organism? It is not the usual hillclimbing task.

I think this local task is best characterized as a *control task* under conditions of uncertainty. The bacterium interacts in a closed-loop fashion with an environment on which its actions exert only modulatory control. It does not have the ability to place itself instantaneously anywhere in space (as is often assumed in framing more abstract optimization tasks); it does not even have the ability to choose a specific direction of travel. Moreover, it has neither a sufficiently rich sensory repertoire nor a sufficiently elaborate internal representation of its local environment, we may safely assume, to allow it to determine which of its two actions (reversing or not reversing flagellar rotation) is the best at any instant. Probabilistic 'Win-Stay / Lose-Shift' (where winning, staying, losing and shifting are all suitably defined in terms of spatial movement) is a control rule that produces effective hillclimbing behavior, given the kinds of attractant distributions encountered, despite the bacterium's restricted capabilities in directly controlling and sensing its environment.

A free-living cell faces uncertainty due to the complexity of the medium in which it lives and its limited local view. A cell in a multicellular organism faces uncertainty for the same reasons, but in this case, these factors arise in part from the decentralization implied by the distributed nature of the system. Remote components can have only limited information about each other and the overall system of which they are parts. Following Wheeler (1985) we distinguish between two types of uncertainty: *endogenous* uncertainty is uncertainty due to unknown aspects of the behavior of other components in the same system; *exogenous* uncertainty is uncertainty due to unknown aspects of the system's external environment and chance external events. In the examples described below, both kinds of uncertainty are factors, but let us first examine a deceptively simple decision problem involving uncertainty.

5.4 A decision problem under uncertainty

Suppose you are given two coins, which are distinguishable as Coin 1 and Coin 2, and you are to conduct a series of trials, or experiments, in each of which you select one of the coins, flip it, and note the outcome (head or tail). Nothing is known about the coins except that they may not be fair. The object of your task is to select a coin at each step, based on knowledge of the history of previous selections and observations, so as to maximize the expected number of heads over some finite or infinite time interval. Clearly, you would like always to select the coin with the larger probability of turning up heads, but you do not know, in fact, you can never know with certainty, which coin this is. Two factors influence each selection: (1) the desire to use what you already know about the coins to obtain immediate payoff, and (2) the desire to acquire more knowledge about the coins in order to make better selections in the future. More generally, the first factor is the desire to *control* the environment based on the current level of knowledge in order to improve performance, whereas the second factor is the necessity to *identify* the environment

in order to make better control decisions in the future. These two factors - the need to control and to identify - ordinarily conflict: the best decision according to one is not best according to the other. Identification requires performing experiments designed to probe the environment for new information, something that by its very nature requires abandoning short-term performance goals. We can see the control/identify conflict in the task faced by the chemotactic bacterium. The bacterium's movement must not only bring it toward higher attractant concentrations but must also serve to detect the gradient of the attractant through the comparison of successively sensed attractant levels.

The decision problem just described in terms of two coins is probably the simplest example showing the control/identify conflict. It is known as the 'two-armed bandit' problem, studied first by Thompson (1933). It is a problem involving the sequential allocation of experiments and has both theoretical and practical importance (the coins, for example, may be replaced by two clinical treatments and the trials by patients). Berry & Fristedt (1985) provide a comprehensive treatment of this subject and a useful annotated bibliography. This class of decision problem is of interest here because the hypothetical neuronal learning rule we explore is a direct extension of one type of strategy that has been applied to these tasks.

5.5 Stochastic learning automata

Of the several theoretical traditions that have developed around the problem of the sequential allocation of experiments, I focus on the theory of learning automata, which originated in the work of the Soviet cybernetician M.L. Tsetlin (1973) and is currently being pursued by engineers (Narendra & Thathachar, 1974), where stochastic learning automaton algorithms are the primary subjects of study. Similar algorithms were independently developed by mathematical psychologists (e.g. Estes, 1950, Bush & Mosteller, 1955). Figure 5.2 shows a stochastic learning automaton interacting with a random environment. At each time step in the processing, the automaton randomly selects an action from a set of possible actions $\{a_1, \ldots, a_n\}$, where a_i is chosen with probability p_i. The environment then evaluates the action and transmits a payoff back to the automaton. For simplicity, we consider only the case in which the payoff is either 'success' or 'failure', but the theory extends to the case of a range of payoff values. The environment determines the payoff according to a set of probabilities $\{d_1, \ldots, d_n\}$, where d_i is the probability of delivering success given that action a_i was selected. Upon receiving the payoff, the automaton updates its action probabilities depending on the action chosen, its current action probabilities, and the payoff received.

Rules for updating action probabilities increase the probability of the action chosen if the payoff indicates success, and decrease it if the payoff indicates failure. The other action probabilities are adjusted so that the new probabilities still sum to one. The magnitudes of these probability changes as functions of the current probabilities are critical in determining the performance of the update rule. Beginning with no knowledge of the success probabilities, the objective of the

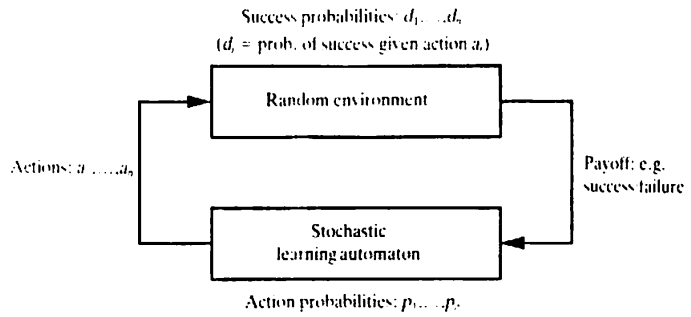

Figure 5.2 Stochastic learning automaton interacting with a random environment.

automaton is to improve its expectation of success over time. Ideally, it should eventually choose, with probability 1.0, the action corresponding to the largest success probability. Many different algorithms have been studied under a number of different performance measures, and many convergence results have been proven (Narendra & Thathacher, 1974).

It is important to note that although the action probabilities, p_i, $i = 1, \ldots, n$, must sum to one, the success probabilities, d_i, do not have to because they are the probabilities of success conditional on the action selected. For example, consider the case of two actions - the case corresponding to the 'two-armed bandit' problem where d_i is the probability of Coin i coming up heads. Each point in the unit square 'contingency space' shown in Figure 5.3 represents a pair of head probabilities, (d_1, d_2), for two possible coins. Assuming that $d_1 > d_2$, so that Coin 1 is the best, we can restrict attention to the lower triangle. If we knew, *a priori*, that the point corresponding to the task being faced falls within a restricted region of contingency space, we could take advantage of this knowledge in devising an algorithm.

For example, the point (.6, .4) corresponds to a task in which the probabilities happen to sum to one. This task is relatively easy because the best action (select Coin 1) tends to yield success more than half the time, whereas the other action tends to yield success less than half the time. If we knew that the success probabilities summed to one, we could devise a simple algorithm that would do well; in fact, we could determine (in the limit) the best action without ever performing one of the actions. The difficulty is that we would like an algorithm that converges to the best action for tasks falling anywhere within contingency space. It turns out that this is not so easy, and in general it is necessary to continue to perform *both* actions with non-zero probability in order to counter the control/identification dilemma.

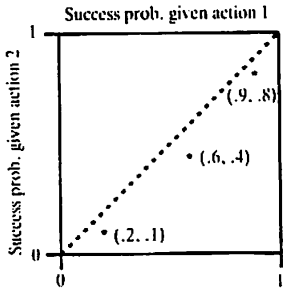

Figure 5.3 Contingency space for the two-action case.

Consider, for example, the tasks corresponding to the points (.9, .8) and (.2, .1) in the contingency space of Figure 5.3. In the first task, both actions yield success with a high probability, making it difficult to decide which is the best. Unsophisticated algorithms often converge to the inferior (but still good) action. In the second task, failure usually results no matter what action is chosen. Unsophisticated algorithms tend to oscillate under these conditions. Fortunately, there are fairly simple stochastic learning automaton algorithms that are able to approach optimal performance for arbitrary contingencies. Here I omit details, which involve subtle issues of stochastic convergence, and simply call such learning automata *competent*. That such learning automata choose their actions probabilistically contributes importantly to their competency. The hypothetical neuronal learning rule we have studied uses a probabilistic firing mechanism because it is an elaboration of a competent stochastic learning automaton.

I would like to make two additional observations about learning automata before turning to their collective behavior. First, the 'Win-Stay / Lose-Shift' strategy in its deterministic form is the first learning automaton studied (Robbins, 1952; Tsetlin, 1973). It is an example of a learning automaton and has been shown to perform better than a totally random strategy for selecting actions, but is far from optimal in these types of decision tasks. Note that the deterministic 'Win-Stay / Lose-Shift' strategy is similar in some respects to an error-correction neuronal learning rule, such as the perceptron rule, in that such a rule only changes weights upon error. In contrast, the rules we consider here change weights more upon success than they do upon failure, a feature essential for their performance in uncertain environments.

A final observation about the learning automaton formalism concerns the meaning of the 'environment.' This is an abstract formalism designed to allow certain kinds of theoretical questions to be framed precisely, and one should not underestimate the degree of its abstraction when relating the formalism to biological systems. In particular, Figure 5.2 shows the environment directly computing the payoff and the learning automaton having a specialized input for receiving it.

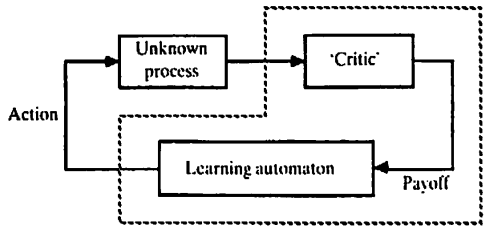

Figure 5.4 An alternative way to view the interaction between a learning automaton and its environment.

However, this is an abstraction that can be implemented in many ways, including that shown in Figure 5.4 in which a learning automaton and a 'critic' (which may itself be adaptive as described by Barto *et al.*, 1983), comprise a learning system (within the dotted lines) that does not receive specialized payoff signals from its environment. In the description of the neuron-like unit given below, the unit has a specialized input pathway for receiving payoff because this seems to be the simplest way to implement the learning rule. However, one can consider units to which payoff is effectively delivered via many pathways, which may also carry other kinds of information, by regarding the unit as preferring some input patterns over others. This kind of implicit payoff may be a more appropriate model for some purposes.

5.6 Collective behavior of learning automata

The random environment shown in Figure 5.2 is an extremely general model of an environment that behaves unpredictably. The ability of a competent learning automaton to improve performance in any such environment serves that automaton well when it has to interact with an environment containing other learning automata. This is the basis for understanding the behavior of collections of learning automata. Figure 5.5 shows a collection of N learning automata interacting with a common environment. The payoff to each automaton depends on the actions of the other automata in addition to its own action. Two different cases are usually studied. In the simplest, each automaton receives the same payoff. This is known as the team problem, or more technically, the 'decentralized team problem with incomplete information' (Narendra & Thathachar, 1974). In a game problem, on the other hand, each automaton receives a different payoff. In this case there may be conflicts of interest among the automata, and the question of what constitutes optimal collective behavior involves the difficult questions studied by game theorists.

A variety of theoretical results exist about the performance of learning automata in team and game problems. In team problems, competent learning

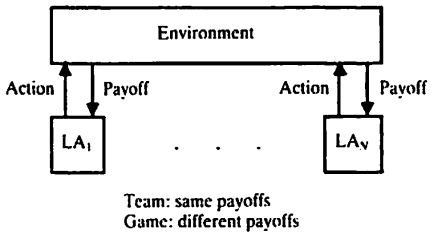

Figure 5.5 A collection of N learning automata interacting with a common environment.

automata have been shown to monotonically improve their performance (Narendra & Wheeler, 1983). In the case of zero-sum games (games of pure conflict), competent learning automata converge to the game's solution in an appropriate probabilistic sense (Narendra & Thathachar, 1974). To illustrate team and game tasks, I briefly describe two examples that Tsetlin presented in a 1965 lecture to physiologists (Tsetlin, 1973).

The first example is the so-called Goore game (although it is a team problem) which I describe as Tsetlin did in terms of human players. Suppose there is a referee and a number of players. The referee can see the players but the players cannot see one another. At the sound of a buzzer, each player is to raise one or two hands. The referee determines what percentage of players raised one hand, then pays each player a fixed amount with a probability that depends on this percentage (for example, according to the functional relationship shown in Figure 5.6). The process repeats each time the buzzer sounds. It can be shown that for any number of players implementing a competent learning automaton strategy, the process converges so that eventually the payoff probability is maximized. The result requires a payoff probability function that has a single maximum, such as that shown in Figure 5.6. Using this particular function, eventually 20% of the players will raise one hand at the sound of the buzzer.

In a Goore game, then, each learning automaton (player) receives the same payoff which is determined by the total number of learning automata that performed the designated action. This is a special case of a team problem where, more generally, the payoff to all the automata can depend on the *pattern* of automaton actions and not just their count. From the perspective of any individual learning automaton, the task is to try to maximize individual payoff in the face of the exogenous uncertainty produced by the probabilistic payoff procedure (Figure 5.6) and the endogenous uncertainty produced by the activity of the other automata - activity that is not directly observed. As far as any of the players is concerned, there are no other players, only a noisy, and non-stationary, environment.

One of the reasons Tsetlin was interested in the Goore game was its potential relevance to the process by which motor units are recruited. If the collection of players is thought of as a motor neuron pool, then the learning process is capable of

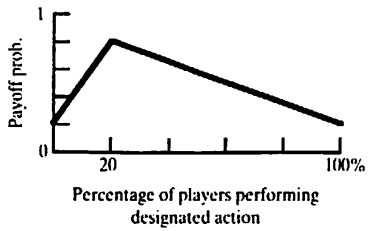

Figure 5.6 Payoff probability as a function of collective action in the Goore game (after Tsetlin, 1973).

adjusting the number of motor neurons that fire. What is perhaps most interesting about this is that, as Tsetlin points out, the payoff does not have to be determined based on the count of the number of neurons firing; it can depend on, say, the total force generated by the motor units, or perhaps on more distal consequences of motor unit activity. The learning process would still produce the required number of active motor neurons (assuming the team does not get stuck in a local optimum). Thus, the collection of learning automata can discover how to achieve some target configuration without requiring an agent in its environment that already knows how the target can be achieved, as would be required if only supervised learning were at work.

Let us look briefly at another example of collective behavior, again following Tsetlin (1973), by considering what he called the distribution game, which, unlike the Goore game, involves players that receive different payoffs. Imagine a collection of animals, each of which feeds from one of a set of feeding troughs. Each trough contains a certain amount of food. At a signal, each animal selects a feeding trough without any form of communication with the other animals. The amount of food each animal receives at each trial depends on the number of animals that have also chosen that trough at that trial. We assume the total amount of food in a trough is divided equally among the animals that approach it. The troughs are then refilled with the same amount of food as before, the signal occurs again, and the process repeats for a sequence of trials. Unlike the team situation, here the players receive different amounts of payoff and can have conflicts of interest. However, to the players, the situation appears identical to that of the Goore game, and competent learning automata learn to distribute themselves in a way that is 'just as reasonable as for people who would know the contents of each trough' (see Tsetlin, 1973, pp. 115-118).

Why do learning automata have to be competent in order to operate effectively as team or game players? Recall that a competent learning automaton is one that can learn which of its actions is best in the absence of *a priori* restrictions on the contingencies it faces. When a learning automaton's environment in part consists of other learning automata, then it faces contingencies that are constantly changing as the other automata are learning (a non-stationary environment). There is no

guarantee, therefore, that the contingencies faced by any of the automata in the collection will remain in any pre-designated region of contingency space. If an automaton happens to be confronted with particularly easy contingencies, it will learn quickly; confronted with difficult contingencies, it will learn more slowly but will not prematurely settle on the wrong action. Learning automata that are not sufficiently competent, such as deterministic learning automata, will sometimes learn to participate in appropriate collective behavior and sometimes not.

In order to apply learning automata to the tasks just described - the Goore game and the distribution game - it had to be assumed that the players were not able to communicate directly with one another. This is because the learning automaton formalism (Figure 5.2) does not have provision for input to the automaton other than the payoff signal. Tsetlin made the following comment:

> 'We have discussed very simple forms of behavior, and for this reason we limited ourselves to the simplest types of automata. The exchanges of information among these automata takes place in the language of penalties and rewards. Although this language seems universal enough, it would, however, be interesting to also look at more complicated automata that possess some specialized language to communicate with other automata. Such automata are needed to describe more complex forms of behavior. These more complex behavioral forms necessitate the use of much more diverse information.' (Tsetlin, 1973, p. 125.)

Our research has focussed on extending the theory of learning automata in the direction Tsetlin suggested in this quoted passage. We now consider more elaborate learning automata that are more like neurons because they make decisions conditionally on input signals which provide information about the state of their environments.

5.7 Associative learning automata

Figure 5.7 shows a learning automaton receiving information, which we call context input, in addition to the payoff signal from its environment (cf. Figure 5.4). The bold arrows in Figure 5.7 are intended to suggest that those pathways potentially transmit massive amounts of data; that is, they represent vector rather than scalar signals. The associative learning automaton must learn how to act conditionally on the context input in order to maximize its expected payoff. Whereas the learning automata described above have to learn a *single* optimal action (or perhaps a single optimal probability for each action), an associative learning automaton tries to learn a *rule*, or mapping, associating context input with optimal actions. What action is best depends on the contingencies currently being implemented by the environment, and usually an environment can provide information that can be associated with appropriate actions. One can think of the context input as providing a clue as to the state of the environment. Context input

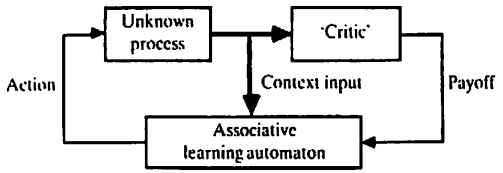

Figure 5.7 A modified version of Figure 5.4 showing an associative learning automaton receiving context input in addition to payoff input.

therefore plays a role similar to that of discriminative stimuli in instrumental conditioning.

Associative learning automata face two kinds of problems. First, they have to learn to classify context input patterns into classes, where all patterns in a given class signal that the same action is best when performed in the presence of any of those patterns. Second, they have to discover what these best actions are. The first problem corresponds to the pattern classification tasks for which a variety of learning rules for neuron-like units have been developed, e.g. the perceptron rule (Rosenblatt, 1961) or Widrow & Hoff's (1960) Least-Mean-Square (LMS) rule. Here, however, there is no teacher able to directly specify desired responses. The learning system has to discover what the best actions are by probing its environment in the manner of a learning automaton.

We have developed numerous algorithms for associative learning automata, but here I focus on the most successful one: the *Associative Reward-Penalty*, or AR-P, algorithm, first introduced by Barto & Anandan (1985) and extensively discussed by Barto (1985). It is a hybrid of perceptron/LMS and stochastic learning automaton algorithms that turns out to be closely related to Thorndike's (1911) Law of Effect and to the 'selective bootstrap adaptation' rule of Widrow *et al.* (1973). The AR-P algorithm applies to a neuron-like unit with a number of input pathways for context input and a specialized input pathway for payoff. I describe its behavior informally; details can be found in the references.

The unit has binary output, its two actions being 'firing' and 'not firing'. As is usual, there is a weight for each context input pathway, and the unit's output is determined by the weighted sum of the context input signals. However, the output depends in a random way on the weighted sum: the more positive the sum, the more likely the unit is to fire; the more negative, the less likely it is to fire (this input/output behavior is identical to that of the Boltzmann units of Ackley *et al.*, 1985). Thus, the AR-P unit is like a two-action stochastic learning automaton except that its action probabilities depend on the context input in a manner determined by the synaptic weights. If an action is followed by feedback indicating 'success', the weights are changed so as to make that action more likely in the presence of the context input pattern in which that action was taken (and patterns similar to it). If an action is followed by 'failure', the weights are changed to make that action less

likely. The role of the random action-generation process is the same as it is for a stochastic learning automaton: to generate sufficient variety in the unit's behavior so that the identification aspect of the unit's task is accomplished effectively.

Note that it is not the synaptic weights that we vary randomly but the unit's firing activity. It would be quite possible to consider a system in which each synapse were represented by a separate (non-associative) stochastic learning automaton, but then the learning process for a single unit would require solving a team decision problem. Learning in this case would be far less efficient than in the one we are considering, in which knowledge is used about how the individual input signals are combined by the unit. Indeed, it would even be possible, and perhaps interesting, to consider each synapse as containing a team of non-associative learning automata, each representing a neurotransmitter vesicle or receptor engaged in a kind of Goore game. Relying solely on this mechanism would be inefficient in the extreme, but learning automata operating at all these levels and facing locally-defined payoff structures could yield very robust adaptive capabilities.

We see, then, that an AR-P unit in effect just implements the Law of Effect, but the weights have to be updated so that the unit is competent (see Section 5.5) in all of the contexts to which it is exposed, or at least, is competent within the confines of its ability to discriminate contexts. An AR-P unit achieves this (but under the restriction that the context input patterns are linearly independent; see Barto & Anandan, 1985). In addition to its probabilistic action-generation process, a feature of the AR-P unit that is critical for its competency is an asymmetry between the magnitudes of the weight changes that occur in the 'success' and 'failure' cases. A much smaller step size must be used in the case of failure; indeed, as the weight changes made upon failure get smaller, the asymptotic behavior of the unit improves. Competency seems to require updating weights in a manner strikingly different from that of an error-correction rule like the perceptron rule, which changes weights only when its response is incorrect. It may be of more than passing interest that the asymmetry between the success and failure cases, required for an AR-P unit to be competent, exactly corresponds to the change in Thorndike's first symmetrical version of the Law of Effect that he adopted to bring his theory into closer accord with experimental observations (see, for example, Hilgard, 1956).

My colleagues and I have written extensively on the differences between associative learning automata and the more familiar error-correction rules for supervised learning commonly used in artificial neural networks (Barto *et al.*, 1983; Barto, 1985). I limit my remarks here to a few observations about differences between the tasks each is capable of solving. As usually formulated, supervised learning, where a teacher directly specifies the desired responses, does not involve any form of feedback through the environment. The environment simply repeatedly presents input patterns paired with desired responses. This is a form of open-loop learning akin to Pavlovian conditioning. There is feedback involved in error-correction rules, of course, but it is feedback of the unit's own response, which can take place within the unit.

In an alternative formulation of error-correction rules for supervised learning that places the error calculation outside of the unit, there *is* feedback passing through the unit's environment, but it is very different from the evaluative feedback under which associative learning automata operate. According to this formulation, learning occurs under the control of error signals that tell the unit *how* to change its actions, whereas evaluative feedback signals do not do this. In supervised learning, a positive error tells the unit that its response was too high, a negative error tells it that its response was too low, and an error of zero signals the desired state of affairs. On the other hand, in the kind of learning we are considering here, the payoff would be 'failure' for both the cases of positive and negative error, and the learning system would have to make adjustments so as to decrease its tendency to do whatever it did, not just decrease or increase its response as specified by the sign of the error. Moreover, if the environment is noisy, then the adjustments must be made carefully to ensure adequate testing of alternatives. Of course, the payoff signal need not be derived from a signed error in this manner (no error = success, plus and minus error = failure). It can be generated by a critic (Figure 5.7) that knows neither what the actual response was nor what it should have been. A critic can generate a payoff based on knowledge solely of *what* it wants accomplished and not of *how* the learning system can accomplish it.

It should be clear that the kind of learning we are considering can be applied to supervised-learning tasks (Barto & Jordan, 1987), but it applies to more difficult tasks as well. However, the usual error-correction rules for supervised-learning are not competent in the kinds of tasks we are considering here. These tasks involve genuine feedback paths through the learning system's environment as in instrumental conditioning experiments. Describing error-correction as 'trial-and-error' learning, as has often been done, is incorrect and misleading. Trial-and-error really means 'generate-and-test,' which only applies superficially to supervised-learning methods. Learning automata, associative or otherwise, are engaged in learning to *control* their environments, not just to mimic them. Learning to mimic environmental events may be an important process in the construction of mental representations of the environment, but it is not the only process of interest.

Is it plausible that a single neuron could implement a competent associative learning automaton strategy? We can only speculate about possible cellular mechanisms, but Koshland's model of bacterial chemotaxis described in Section 5.2 suggests a starting point. One of the requirements of such a mechanism, if it is to operate in real time, is that it must possess a form of short-term memory to retain a trace of context inputs, and the actions taken in their presence, for sufficient time to allow the return of evaluative feedback. This requires a mechanism more complex than Koshland postulated for bacterial chemotaxis (Figure 5.1), but that model suggests how this short-term memory could be implemented within a cell. We could, for example, replace the tumble regulator X by a randomly varying membrane potential or firing threshold, allowing the firing probability to be conditional on synaptic input. Attractants and repellents would correspond to various neuromodulators delivered via diffuse projections from brain reinforcement centers and from sources that are more local. The detection of changes in the

concentrations of these substances could be accomplished as in Koshland's model, and traces of relevant past information could be maintained by similar biochemical means as long as the time interval is not too long (perhaps hundreds of milliseconds to a few seconds). Traces of the recent past history of synaptic activity would have to be maintained in some synaptically-local manner. Sutton & Barto (1981) discuss hypothetical mechanisms for storing these kinds of traces at synapses, and more recent knowledge about cellular mechanisms suggest others (see, for example, Alkon & Rasmussen, 1988). Some of the theoretical issues involving time delays between actions and their consequences have been studied, but this is outside the scope of the present chapter (see Sutton & Barto, 1981; Barto *et al.*, 1983; Sutton, in press).

5.8 Collective behavior of associative learning automata

The behavior of collections of associative learning automata can be more complex than the behavior of the collections of non-associative automata considered above. I present two examples that suggest some of the possibilities. The first example is a team decision problem illustrating some of the important features of decentralized decision making that are absent in the non-associative case, and which I believe may be important in the functioning of the brain. The second example shows how the problem of learning in a layered network can be considered as an extension of the decentralized team decision task, where part of the context input to a unit is determined by the actions of other units.

5.8.1 *Decentralized decision making*

In a tutorial on decentralized statistical decision making, Yu-Chi Ho (1980) used a simple example that emphasizes the 'essentials of team theory by stripping away all unimportant details'. I describe the task just as he did, and then I show how AR-P units learn how to act as effective decision makers in this task. Mr. B, who lives in Boston, and Mr. H, who lives in Hartford (about 100 miles away), need to meet in Worcester, about midway between Boston and Hartford, in order to close a business deal. They decide to meet in Worcester the next day at noon if it doesn't rain, but are unable to communicate further. As Ho says, 'New England weather being what it is, an *uncertainty* has developed about whether or not it is raining in Worcester when H and B are about to depart for their meeting. Of course, each of them has access to his *own* local weather information in his city. This information is in turn correlated with the state of the weather in Worcester as well as with the information received by the other person.' Assuming that both H and B have to be present to conduct their business and that sunshine in Worcester is essential to its successful outcome, Ho postulates the payoff matrices shown in Figure 5.8.

Rain in Worcester

Mr. B \ Mr. H	go	don't go
go	-4	-2
don't go	-2	5

Shine in Worcester

Mr. B \ Mr. H	go	don't go
go	10	-3
don't go	-3	0

Figure 5.8 Payoff matrices for team decision problem (after Ho, 1980, Figure 1).

Mr. B and Mr. H have to decide whether or not to make the trip to Worcester. More than this, however, we require them to come up with a strategy for deciding when to make the trip depending on their local weather conditions. According to Ho (1980) this problem has the main features of team decision problems (quoting Ho):

1) the presence of *different but correlated* information for each decision maker about some underlying *uncertainty*;
2) the need for *coordinated* actions on the part of all decision makers in order to realize the *payoff*.

If the problem does not have both of these features, then it simplifies. For example, if the cities involved are so far apart that their weather is uncorrelated, then it is does not help to consider local weather conditions; if the tasks to be performed by the decision makers do not require coordination, then the decisions can be made independently.

In the example task, each decision maker can adopt one of four possible decision strategies (different mappings from the two possible local weather conditions to the two possible actions). Thus, there are sixteen possible strategies for the two players, and which one is best depends on the correlations among the weather conditions in the two cities. These correlations arise from the joint probability distribution of the three random variables describing the weather in the three cities. Given such a distribution function, it is possible to determine which of the sixteen strategies yield maximal expected payoff. Figure 5.9 is one such joint distribution given by Ho, and with some computation one can show that the optimal strategy is for both Mr. B and Mr. H to ignore local weather conditions and always travel to Worcester (see Ho, 1980, for details and more complex examples).

This is exactly the kind of problem that associative learning automata, such as AR-P units, should be able to solve. We investigated this by letting two AR-P units play the roles of the decision makers as shown in Figure 5.10. A unit firing means that the decision to make the trip has been made. At each time step, we selected a weather pattern for the three cities according to a given joint probability distribution. The weather selected for Boston was coded as a 0 or a 1 and given as context input to the unit representing the Boston decision maker. Similarly, Hartford's weather provided context input to the other unit. The weather selected for Worcester

Boston	r	r	r	r	s	s	s	s
Hartford	r	r	s	s	r	r	s	s
Worcester	r	s	r	s	r	s	r	s
Prob.	0.25	0.05	0.1	0.1	0.1	0.1	0.05	0.25

Figure 5.9 Joint probability distribution for the weather in Boston, Hartford, and Worcester (r = rain, s = shine; after Ho, 1980, Figure 2).

determined which payoff matrix of Figure 5.8 we used to determine the payoff as a function of the units' actions. We also provided each unit with a variable bias, in the form of a constant input signal multiplied by an adjustable weight, so that it could, if necessary, learn to fire with a high probability when being presented with a context signal of 0. Our simulation experiments suggest that AR-P units are consistently able to find the best strategies for a variety of payoff matrices and weather pattern probabilities, including those shown in Figures 5.8 & 5.9, for which both units learned to stay on.

Despite being able to make decisions conditionally upon context information, the associative learning automata shown in Figure 5.10 have no direct means for communicating with one another. They are therefore unable to agree among themselves upon some course of action *before* they act. This means that this example, as implemented, is a *non-cooperative* problem: no pre-game communication and agreement is possible. The players come to coordinate their actions as a result of the learning process, but no direct coordination is permitted. In contrast, the theory that Ho (1980) discusses is the fully cooperative case in which the players are assumed to be able to communicate with each other when they are designing their strategies, and because he is not studying learning, he does not consider repeated play of the game. He is concerned with the problem of determining what constitutes a good solution to a decentralized decision problem given complete knowledge of the payoff structure and other characteristics of the environment. We have been concerned, on the other hand, with the non-cooperative case and incomplete information, and have focussed on learning. What we can consider within our framework, however, is how decision makers can *learn to cooperate* with one another. I turn now to an example showing how competent associative learning automata can learn how to coordinate their actions for mutual benefit. This takes the form, by now familiar to neural-network researchers, of a layered network.

5.8.2 Layered networks

In previous publications, my colleagues and I described simulation experiments showing how layered networks of AR-P units can learn to solve non-

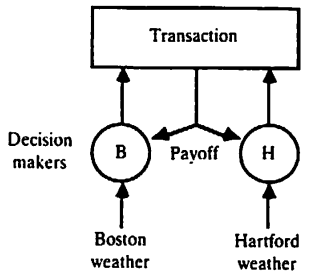

Figure 5.10 Two associative learning automata in a decentralized decision task.

linear associative learning tasks (Barto, 1985; Barto *et al.*, 1986; Barto & Anderson, 1985; Barto, 1986; Barto & Jordan, 1987). Here I describe the simplest of these experiments as an example of a distributed problem solving task in which units learn how to cooperate. Figure 5.11 shows a network of two associative learning automata, labelled A and B, that use the AR-P algorithm. The context input to Unit A provides environmental state information as a stimulus from the environment, and the context input to Unit B tells it which action Unit A has selected (as before, each unit also has a constant input with an adjustable weight so that it can learn to turn on in the presence of a zero as context input). Thus, only Unit A receives input from the environment external to the network, and only the action of Unit B directly influences the payoff. Both units receive the same payoff, making it a team problem.

Suppose we place this network in the following discrimination task. The environment has two states, one of which is selected at random on each trial. When in one state, State 1, the environment responds to Unit B's firing by providing a payoff of 'success' to both units with high probability and 'failure' with low probability; whereas if Unit B does not fire, the State 1 environment responds with 'failure' to both units with high probability and 'success' with low probability. When the environment is in State 0, on the other hand, the contingencies are reversed so that it provides 'success' with the highest probability if Unit B does not fire. The environment signals its state to Unit A by sending a 1 or a 0 to Unit A via its context inputs. Therefore, in order to maximize the expectation of success, the network as a whole should simply construct the identity mapping, producing output 1 in response to input 1, and output 0 otherwise.

If there were no communication link between Unit A and Unit B, only limited success rate could be achieved. Unit B, unable to sense a discriminative stimulus, would solve the task non-associatively by learning always to select the single action that is the best to perform independently of the network input. Unit A, unable to influence the environment at all, would never settle on a consistent action because nothing it could do would matter. The complementary specialities of the two units

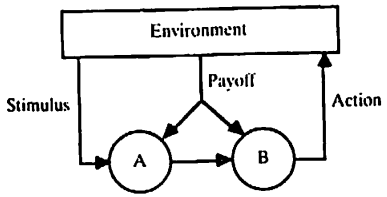

Figure 5.11 A two-layer network of associative learning automata.

have to be combined for the network to achieve an optimal success rate - the units have to cooperate. The interconnecting link from Unit A to Unit B has an adjustable weight that allows Unit A's decision to influence that of Unit B. We set this weight to zero initially in the simulations, and if it is suitably adjusted, then the network can respond correctly to the discriminative stimulus. However, the correct value of the interconnecting weight depends on how Unit A has learned to respond to its input; conversely, the correct behavior of Unit A depends on the value of the interconnecting weight, that is, on how Unit B has learned to respond to its context signal. Thus the two units must adapt simultaneously in a tightly-coupled cooperative fashion. Simulations, more completely discussed in Barto (1985), strongly suggest that the network always converges so as to maximize the expectation of success.

A non-zero value for the weight on the pathway connecting Unit A to Unit B really does mean that the units cooperate, albeit in a limited way. Their activity is no longer statistically independent. To the extent that the weight has large magnitude, there is a *binding agreement* between the units that is acted upon before the network makes an overt action: Unit B responds to what Unit A tells it. Units A and B form a kind of coalition. If the units were reciprocally connected, then the process of agreeing upon an action would take place via an iterative relaxation process, although we have not yet thoroughly studied this case.

Although I have presented this method for learning in multilayer networks in terms of distributed decision making, it has not been my intent to obscure the relationship between this method and other methods for obtaining effective learning in multilayer networks, notably the error back-propagation method (Rumelhart *et al.*, 1986). With a few technical caveats, it appears that layered networks of AR-P units update their interconnection weights in a manner that statistically approximates a gradient of overall network performance as shown by Williams (1986, 1987). Thus, it is not misleading (although not exactly technically correct) to view layered networks of AR-P units, acting as a team, as performing a kind of statistical approximation to the process carried out by error back-propagation.

To gain a better understanding of what happens as layered networks of AR-P units learn, consider a single unit in the interior of a layered network (i.e. a 'hidden unit'). Suppose this unit can vary its output about its current value while the outputs of all the other units remain fixed. By correlating the variation of its output with the consequences of this variation on the overall performance measure for the network, the unit can obtain a good estimate of how its activity influences this measure, that is, it can estimate the derivative of the performance measure with respect to its output at the current point in weight space. From this the unit can easily determine the performance measure's derivative with respect to its weights, and so can alter its weights appropriately. Suppose each unit does this in turn with the other units' outputs fixed. If a unit's new weights are not put into place until all the units have varied their outputs, the result will be a step in weight space according to a good estimate of the gradient of the performance measure. It turns out that it is possible for interacting units to vary their outputs simultaneously to obtain estimates of the appropriate derivatives. This estimation process is made difficult by the endogenous noise produced by the simultaneous variation, but AR-P units are able to sort out the correct information. Note that it would also be possible to vary simultaneously all the network's *weights* to obtain gradient estimates, but this is not done in AR-P networks and it would be considerably slower.

Because the AR-P method estimates gradient information, whereas the error back-propagation method computes it during the backward pass, the AR-P method is considerably *slower* than error back-propagation, which itself is regarded as being too slow for scaling up to large applications. Nevertheless, several things can be said in defense of networks of AR-P units. First, networks of associative learning automata are not restricted to solving supervised-learning tasks; they can be effective for a wide class of decision-making tasks involving uncertainty (which includes supervised learning as a special case) as I have tried to emphasize in this chapter. Second, as I have also emphasized in this chapter, networks of cooperating associative learning automata have a kind of biological plausibility that, in my opinion, algorithms such as error back-propagation lack. The learning of cooperative collective behavior is a very natural consequence of competent associative learning automata interacting under conditions that make it advantageous for the individual units to coordinate their activity. There is no need for a tightly regimented training procedure.

Finally, that competent associative learning automata are robust enough to improve their performance under rather unstructured conditions suggests that this approach may provide new ways of obtaining efficient learning in modularly organized, perhaps hierarchical, adaptive networks. In all the examples I have presented involving associative learning automata, the payoff signal is diffusely broadcast to all the units. Although there are diffuse projections in the brain that could serve this function for large groups of neurons, learning becomes much more rapid as the number of units to which a payoff signal is uniformly broadcast decreases. Sending different and specialized payoff signals to subgroups of units can vastly increase the speed of learning, but requires more knowledge on the part of whatever is serving as a critic, knowledge which may itself be learned with experience. The *global* broadcast of a single payoff signal should be viewed as a *worst case*. Various forms of local reinforcement systems have to exist for large

problems to be solvable by this kind of learning. One way of achieving this is suggested by the fact that training can be accomplished by agencies that only know *what* they want accomplished, but not *how* it can be accomplished. Modules can cause things to happen that they cannot do themselves by setting appropriate contingencies for other modules having the necessary expertise and access to the necessary information. Analogies with social and economic systems are obvious, although highly structured regimes that would be unacceptable to organisms like ourselves would have to exist at these microscopic levels. These are issues we are exploring in our current research.

5.9 From chemotaxis to cooperativity

In this chapter I have suggested how adaptive strategies that we know are within the capabilities of single cells can be extended to produce a variety of forms of collective behavior, including the learning of cooperative interactions. To mitigate the degree of speculation unavoidable in hypothesizing that neurons, even just some of them, are competent associative learning automata, I have proceeded in small steps leading from bacterial chemotaxis to cooperative collective behavior, and focussed on theoretical questions that have spawned rich theoretical traditions. That such simple ideas, suitably refined, can underlie a wide range of behavior suggests that the hypothesis is worth serious experimental study.

Acknowledgement

This research was supported by the Air Force Office of Scientific Research, Bolling AFB, through grant AFOSR-87-0030. The author wishes to thank Robert Jacobs for his essential help in preparing some of the material presented here and John Moore for valuable feedback on an early draft.

References

Ackley D.H., Hinton G.E. & Sejnowski T.J. (1985) A learning algorithm for Boltzmann Machines. **Cognitive Sci. 9**, pp. 147-169

Alkon D.L. & Rasmussen H. (1988) A spatial-temporal model of cell activation. **Science 239**, pp. 998-1005

Barto A.G. (1985) Learning by statistical cooperation of self-interested neuron-like adaptive elements. **Human Neurobiol. 4**, pp. 229-256

Barto A.G. (1986) Game-theoretic cooperativity in networks of self-interested units. In: **Neural Networks for Computing.** J.S. Denker (ed.) New York: AIP Conference Proceedings 151, American Institute of Physics, pp. 41-46

Barto A.G. & Anandan P. (1985) Pattern recognizing stochastic learning automata. **IEEE Trans. Sys. Man Cybern. 15**, pp. 360-375

Barto A.G., Anandan P. & Anderson C.W. (1986) Cooperativity in networks of pattern recognizing stochastic learning automata. In: **Adaptive and Learning Systems** K.S. Narendra (ed.) Plenum, New York, pp. 235-246

Barto A.G. & Anderson C.W. (1985) Structural learning in connectionist systems. **Proc. Seventh Ann. Conference Cognitive Science Society,** Irvine, CA.

Barto A.G. & Jordan M.I. (1987) Gradient following without back-propagation in layered networks. **Proc. IEEE First Ann. Int. Conference Neural Networks,** IEEE Catalog #87TH0191-7, vol II, pp. 629-636

Barto A.G., Sutton R.S. & Anderson C.W. (1983) Neuronlike adaptive elements that can solve difficult learning control problems. **IEEE Trans. Sys. Man Cybern. 13**, pp. 834-846

Berry D.A. & Fristedt B. (1985) **Bandit Problems: Sequential Allocation of Experiments.** Chapman & Hall, London.

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

Estes W.K. (1950) Toward a statistical theory of learning. **Psych. Rev. 57**, pp. 94-107

Hilgard E.R. (1956) **Theories of Learning.** Appleton-Century-Crofts Inc., New York.

Hinrichsen R.D. & Schultz J.E. (1988) *Paramecium*: a model system for the study of excitable cells. **TINS 11**, pp. 27-32

Ho Yu-Chi (1980) Decision theory and information structures. **Proc. IEEE 68**, pp. 644-654

Kleinrock L. (1985) Distributed systems. **Communications of the ACM 28**, pp. 1200-1213

Klopf A.H. (1972) Brain function and adaptive systems - a heterostatic theory. Air Force Cambridge Research Laboratories Research Report, **AFCRL-72-0164**, Bedford, MA.

Klopf A.H. (1982) **The Hedonistic Neuron: A Theory of Memory, Learning and Intelligence.** Hemisphere, Washington, DC.

Koshland D.E. Jr. (1980) Bacterial chemotaxis in relation to neurobiology, **Ann. Rev. Neurosci. 3**, pp. 43-75

Lackie J.M. (1986) **Cell Movement and Cell Behavior.** Allen & Unwin, London.

Narendra K.S. & Thathachar M.A.L. (1974) Learning automata - a survey. **IEEE Sys. Man Cybern. 4**, pp. 323-334

Narendra K.S. & Wheeler R.M. Jr. (1983) An N-player sequential stochastic game with identical payoffs. **IEEE Trans. Sys. Man Cybern. 13**, pp. 1154-1158

Robbins H. (1952) Some aspects of the sequential design of experiments, **Bull. Am. Math. Soc. 58**, pp. 527-532

Rosenblatt F. (1961) **Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms.** Spartan Books, Washington DC.

Rumelhart D.E., Hinton G.E. & Williams R. J. (1986) Learning internal representations by error propagation. In: **Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol 1: Foundations.** D.E. Rumelhart & J.L. McClelland (eds.) Bradford Books/MIT Press, Cambridge MA.

Selfridge O. (in press) **Tracking and Trailing.** Bradford Books/MIT Press, Cambridge MA.

Sutton R.S. (in press) Learning to predict by the method of temporal differences. **Machine Learning**

Sutton R.S. & Barto A.G. (1981) Toward a modern theory of adaptive networks: expectation and prediction. **Psych. Rev. 88**, pp. 135-171

Thompson W.R. (1933) On the likelihood that one unknown probability exceeds another in view of the evidence from two samples. **Biometrika 25**, pp. 275-294

Thorndike E.L. (1911) **Animal Intelligence.** Hafner, Darien Conn.

Tsetlin M.L. (1973) **Automaton Theory and Modeling of Biological Systems.** Academic Press, New York.

Wheeler R.M. Jr. (1985) **Decentralized Learning in Games and Finite Markov Chains**, PhD Dissertation, Yale University.

Widrow B., Gupta N.K. & Maitra S. (1973) Punish/reward: Learning with a critic in adaptive threshold systems. **IEEE Trans. Sys. Man Cybern. 5**, pp. 455-465

Widrow B. & Hoff M.E. (1960) Adaptive switching circuits, **1960 WESCON Convention Record Part IV**, pp. 96-104

Williams R.J. (1986) Reinforcement learning in connectionist networks: A mathematical analysis. **Technical Report ICS 8605**, Institute for Cognitive Science, University of California at San Diego, La Jolla, CA.

Williams R.J. (1987) Reinforcement learning in connectionist systems. **Technical Report NU-CCS-87-3**, College of Computer Science, Northeastern University, 360 Huntington Avenue, Boston, MA.