Biol. Cybern. 40, 201-211 (1981)

Biological Cybernetics

Associative Search Network: A Reinforcement Learning Associative Memory

Andrew G. Barto, Richard S. Sutton, and Peter S. Brouwer
Department of Computer and Information Science, University of Massachusetts, Amherst, MA 01003, USA

Abstract. An associative memory system is presented which does not require a "teacher" to provide the desired associations. For each input key it conducts a search for the output pattern which optimizes an external payoff or reinforcement signal. The associative search network (ASN) combines pattern recognition and function optimization capabilities in a simple and effective way. We define the associative search problem, discuss conditions under which the associative search network is capable of solving it, and present results from computer simulations. The synthesis of sensory-motor control surfaces is discussed as an example of the associative search problem.

Numerous reports have appeared in the literature describing associative memory systems in which information is distributed across large areas of the physical memory structure (e.g., Amari, 1977; Anderson et al., 1977; Cooper, 1974; Kohonen, 1977; Nakano, 1972; Wigström, 1973; Willeshaw et al., 1969). The simplest of these are based on the properties of correlation matrices, and all of them exhibit interesting and suggestive forms of content addressability, generalization, and error tolerance. There have also been numerous discussions of the possibility that these forms of memory structures may provide models of biological memories. In all of these studies, the storage process is one in which a series of "keys" and "patterns" are repeatedly presented to the memory network which stores the key-pattern associations.

As models of memory, these associative memory structures suggest how a rapprochement might be reached between connectionistic, locationalistic views of memory and Gestalt, mass action views (e.g., Freeman, 1975; John and Schwartz, 1978). Associative memories use learning rules that are connectionistic in character yet need not store information in localized form. However, as models of learning they exhibit only a very simple form of open-loop learning. Since the desired response (the pattern to be reproduced) and the stimulus intended to elicit that response (the key) are both explicitly presented to the system during the training phase, these studies do not address the case of learning in which neither the associative memory nor the environment knows the desired response.

In this paper we describe an associative memory structure which is not told by some outside process (e.g., a "teacher") what pattern it is to associate with each key. Instead, for each key, the network must search for that pattern which maximizes an external payoff or reinforcement signal. As this kind of learning proceeds, each key causes the retrieval of better choices for the pattern to be associated with that key. What gets stored in the associative memory is a result of reinforcement feedback through the environment. By eliminating the need for a "teacher" to explicitly provide the pattern to be stored, the ASN effectively solves a central problem faced by an adaptive system. No part of the system need have a priori knowledge about what associations are best. It is capable of what Widrow et al. (1973) call "learning with a critic". A critic need not to know what each optimal response is in order to provide useful advice.

The ASN combines two types of learning which are usually only considered separately. First, it solves a pattern recognition problem by learning to respond to each key with the appropriate output pattern. This is the problem solved by the associative memory systems described in the literature. The method used is similar to stochastic approximation pattern recognition methods [see, for example, Duda and Hart (1973) for a good discussion of these techniques]. At the same time, the ASN uses a different type of learning to actually find what output pattern is optimal for each key. It effectively performs a search using a stochastic auto-

0340-1200/81/0040/0201/$02.20

maton method to maximize a payoff or reinforcement function. Stochastic automaton search methods originated in the work of Tsetlin (1971) and are reviewed by Narendra and Thathachar (1974). Other systems capable of performing this kind of search do not perform the pattern recognition task. For example, the ALOPEX system of Harth and Tzanakou (1974) to which the ASN is closely related, performs a search but is not sensitive to different input patterns and thus is not an associative memory. The learning the ASN accomplishes solves both the search and the pattern recognition problems in a simple and effective way.

Although learning systems capable of solving both types of problems have been discussed in the adaptative system theory literature (Mendel and McLaren, 1970), these systems do not have the error tolerance and generalization capabilities of distributed associative memories. The only neural theory which contains this synthesis is that of Klopf (1972, 1979, 1981). Klopf emphasizes closed-loop reinforcement learning and correctly points out that, despite common opinion to the contrary, it has been largely neglected by neural theorists. The results presented here demonstrate the significance and novelty of Klopf's theory. We will discuss the ASN in light of Klopf's theory below. Also closely related is the notion of "boot-strap adaptation" of Widrow et al. (1973).

The Associative Search Problem

Figure 1 shows an ASN interacting with an environment E. At each time t, E provides the ASN with a vector $\mathbf{X}(t)=(x_1(t), \ldots, x_n(t))$, where each $x_i(t)$ is a positive real number, together with a real valued payoff or reinforcement signal $z(t)$. The ASN produces an output pattern $\mathbf{Y}(t)=(y_1(t), \ldots, y_m(t))$, where each $y_i(t) \in \{0, 1\}$, which is received by E. The problem the ASN is designed to solve can be stated informally as follows. Each vector $\mathbf{X}(t)$ provides information to the ASN about the condition or state of its environment at time t, or, viewed in another way, provides information about the sensory context in which the ASN should act. We call each $\mathbf{X}(t)$ a *context vector*. Different actions, or output patterns, are appropriate in different contexts. As a consequence of performing an action in a particular context, the ASN receives from its environment, in the form of a payoff or reinforcement signal, an evaluation of the appropriateness of that action in that context. The ASN's task is to act in each context so as to maximize this payoff. By the use of the term context we mean nothing more than the environmental background in which an action is taken, and we do not wish to imply that all of this term's more specialized meanings are applicable here.

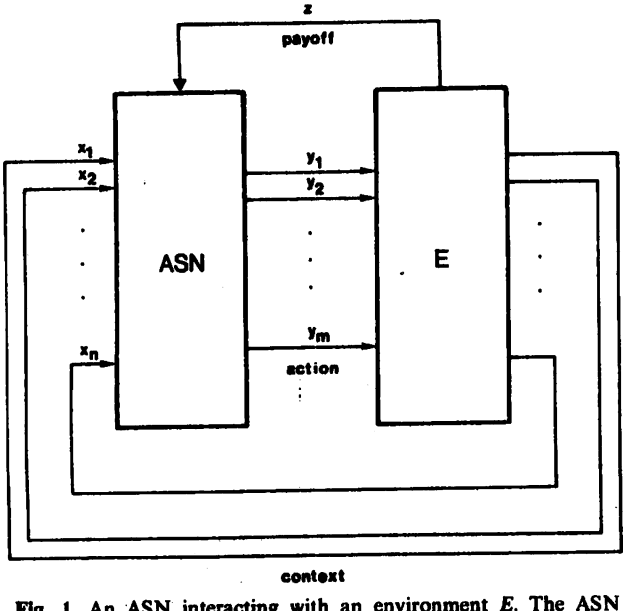

Fig. 1. An ASN interacting with an environment E. The ASN receives context signals $x_1, \ldots, x_n$ and a payoff or reinforcement signal z from E and transmits actions to E via output signals $y_1, \ldots, y_m$

More formally, we assume that $\mathbf{X}(t)$ belongs to a finite set $\mathbf{X}=(\mathbf{X}^1, \ldots, \mathbf{X}^k)$ of context vectors and that to each $\mathbf{X}^\alpha \in \mathbf{X}$ there corresponds a payoff or reinforcement function Z^α. Assuming that E always evaluates an output vector in one time step, if $\mathbf{X}(t)=\mathbf{X}^\alpha$, then $z(t+1)=Z^\alpha(\mathbf{Y}(t))$. We say that E provides a *training sequences over X* if it implements an infinite sequence of payoff functions and emits the corresponding sequence of context vectors

$$\mathbf{X}^{i_1}, \mathbf{X}^{i_2}, \ldots, \mathbf{X}^{i_\nu}, \ldots$$

such that each $\mathbf{X}^{i_\nu} \in \mathbf{X}$ and each element of $\mathbf{X}$ occurs infinitely often (Nilsson, 1965). The associative search problem is solved if, after some finite portion of a training sequence, the ASN responds to each $\mathbf{X}^\alpha \in \mathbf{X}$ with the output pattern $\mathbf{Y}^\alpha=(y_1^\alpha, \ldots, y_m^\alpha)$ which maximizes Z^α. Generalizations of this problem are discussed below.

Although the associative search problem is closely related to the problem that other learning rules, such as the perceptron (Minsky and Papert, 1969; Rosenblatt, 1962), are able to solve, it differs in an important way. The associative search task requires the system to produce output *vectors* based on *scalar* feedback from the environment. An associative memory consisting of perceptrons (see Amari, 1977), on the other hand, would require a separate error feedback from the environment for every component of its output vector. The elimination of the need for the environment to provide such error vectors is, in some regards, equivalent to the elimination of the need for the environment to know each correct system response.

changes. However, when the context vector changes, the change in the value of z is due to the change in payoff function as well as the adaptive element's action. The difficulty this creates can be clearly appreciated by considering the worst case in which the payoff function changes at every time step. Consecutive values of z in this case result from evaluating different functions rather than the same function twice and hence do not provide useful gradient information about any single payoff function. Unless the payoff functions implemented by E vary smoothly over time, one would not expect an adaptive element operating according to (1) and (2) to be capable of solving an associative search problem.

Two methods of solving the problem of context transitions are used in the examples which follow. One is to require E to implement each payoff function, and emit the corresponding context vector, for at least two consecutive time steps and, when transitions do occur, to set the learning constant c to zero so that the change in payoff due to the transition has no effect. This procedure requires either a priori knowledge about when transitions occur or a mechanism for detecting transitions. Such mechanisms can be devised [Didday (1976) and Grossberg (1976) discuss this problem and propose neurally plausible methods]. For simplicity in some of the examples to follow we set c to zero "manually" when a transition occurs.

In other examples, however, we use a method that does not require transitions to be known or detected. Suppose the adaptive element produced action $y(t-1)$ in response to context vector $\mathbf{X}(t-1)$. Instead of comparing the resulting payoff $z(t)$ with $z(t-1)$ which may have been determined by a different payoff function, we compare it with the payoff "expected" for acting in context $\mathbf{X}(t-1)$. If a higher than expected value is obtained, then the action which produced it is made more likely to occur in that context again. In this way, the gradient of each payoff function can be estimated from samples which do not occur consecutively in time. Instead of computing weight values according to (2), we use the following rule:

$$w_i(t+1)=w_i(t)+c[z(t)-p(t-1)]\cdot[y(t-1)-y(t-2)]x_i(t-1)$$

which differs from (2) by the substitution for $z(t-1)$ the value $p(t-1)$ predicted for $z(t)$ given $\mathbf{X}(t-1)$.

We use another type of adaptive element to compute $p(t-1)$ from $X(t-1)$. This element is a variant of one described previously in Sutton and Barto (1981), and proposed as a model of classical conditioning. It learns to *anticipate* the payoff rather than to maximize it, and we call it a predictor. The predictor has n context pathways x_i, $i=1,\ldots,n$, one payoff pathway z, and one output pathway p. Associated with each context pathway x_i is a variable weight wp_i. The output at time t is

$$p(t)=\sum_{i=1}^{n} wp_i(t)w_i(t).$$

The weights change over time according to the following equation: for $i=1,\ldots,n$,

$$wp_i(t+1)=wp_i(t)+cp[z(t)-p(t-1)]x_i(t-1),$$

where cp is a learning constant determining the rate of learning. This element implements a stochastic approximation method for finding weights (if such weights exist) such that $p(t-1)=z(t)$ for all t. If a linear prediction is not possible, the predictor will find the best-least-square linear prediction if cp is allowed to decrease over time. See Duda and Hart (1973) and Kasyap et al. (1970) for good discussions of these methods.

A Network

Figure 3 shows an ASN consisting of m adaptive elements and one predictor. Each context pathway from the environment connects to each adaptive element and to the predictor, as does the payoff pathway z. The adaptive element weights form an $m \times n$ matrix $\mathbf{W}=(w_{ij})$ where w_{ij} is the weight of the i-th adaptive element for the j-th context pathway. The random variables NOISE for each element are independent and identically distributed, and the learning constants are the same for each element.

While the training sequence is being presented, each adaptive element comprising the ASN faces the problem discussed above of maximizing each payoff function. Each element's payoff appears to have a random component since it depends on the unknown outputs of the other adaptive elements comprising the ASN. As a result of the capability of each adaptive element to increase its expected payoff when interacting with an environment having random response characteristics, an ASN consisting of any number of adaptive elements can solve the corresponding associative search problem under certain conditions.

For each context vector, the ASN search problem is an example of what is known in the theory of stochastic automata as a cooperative game of learning automata (Narendra and Thathachar, 1974). Unlike other learning automata studied, however, the ASN solves such a problem for each context vector. By combining notions from the theory of cooperative games of learning automata and the theory of pattern recognition, we can formulate a conjecture about the conditions under which the ASN as described here can

The Basic Adaptive Element

An ASN consists of a number of identical adaptive elements each determining a component of the system's actions. It is useful to describe first a single element which can be regarded as the simplest ASN ($m=1$). Figure 2 shows an adaptive element interacting with an environment E. The element has n context input pathways x_i, $i=1,\ldots,n$, one payoff or reinforcement pathway z, and one output y. Associated with each context pathway x_i is a real valued weight w_i with value $w_i(t)$ at time t. Let $\mathbf{W}(t)$ denote the weight vector at time t. Let $s(t)$ denote the weighted sum at time t of the context inputs. That is,

$$s(t)=\sum_{i=1}^{n} w_i(t)x_i(t)=\mathbf{W}(t)\cdot\mathbf{X}(t).$$

The output $y(t)$ is determined from $s(t)$ as follows:

$$y(t)=\begin{cases}1 & \text{if } s(t)+\text{NOISE}(t)>0\\ 0 & \text{otherwise,}\end{cases} \tag{1}$$

where NOISE is a random variable with mean zero normal distribution. The sum s therefore *biases* the element's output (cf. Harth and Tzanakou, 1974): positive s making it more likely to be 1 and negative s making it more likely to be 0.

The weights w_i, $i=1,\ldots,n$, change according to a discrete time iterative process. At each time step, each weight is updated according to the following equation: for $i=1,\ldots,n$,

$$w_i(t+1)=w_i(t)+c[z(t)-z(t-1)]\cdot[y(t-1)-y(t-2)]x_i(t-1), \tag{2}$$

where c is a constant determining the rate of learning. Other rules also work, but this is one of the simplest. Also for simplicity the response latency for the element is zero; that is, there is no delay between input and output. This causes no difficulties here because we do not consider recurrent connections within a network. In other variants, the inputs need not be positive, and the noise need not be normally distributed. If the context term $x_i(t-1)$ were removed from (2), the resulting learning rule would be essentially that used by Harth and his colleagues in the ALOPEX system (Harth and Tzanakou, 1974).

To understand how (2) works, consider a simple example. Suppose a positive context signal was present on pathway x_i at some time $t-1$, signalling some condition of the environment. Suppose also that $y(t-1)=1$ while $y(t-2)=0$ (that is, the element "turned on" at time $t-1$), perhaps due to an excitatory effect of signal x_i or perhaps by chance. Then, if the payoff signal z increases from time $t-1$ to t (possibly as a result of the element's action), w_i will increase.

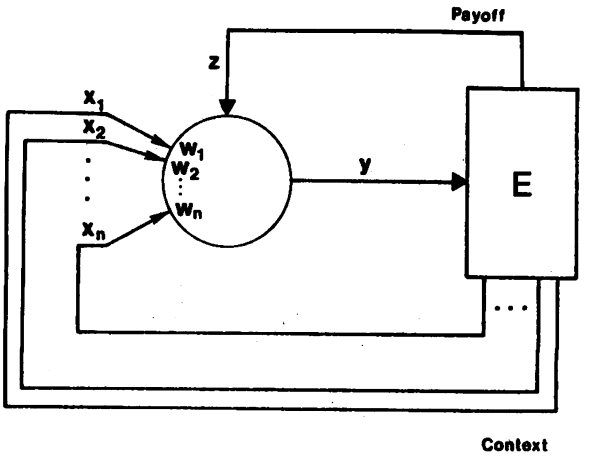

Fig. 2. The simplest ASN: A single adaptive element interacting with and environment E

Since $w_i(t)x_i(t)$ is used to compute $y(t)$, the increased weight w_i will make it more likely (other things being equal) that y will be 1 when signal x_i occurs in the future. Similarly, if z decreases following the element's action, w_i will decrease thereby decreasing the probability that y will be 1 when signal x_i occurs again. Consequently, if turning on in a specific context is followed by an increase in payoff, the element will be more likely to turn on (or stay on) in that context in the future. Other cases can be analyzed similarly: if going off in a context leads to a payoff increase, then the probability of being off in that context increases. Of course, a pathway can participate in signalling a large number of different contexts. This is where the associative memory properties become relevant.

For an ASN consisting of a single adaptive element, the search for the optimal action for each context vector is not very difficult since the ASN has only two actions. However, a property of the adaptive element that is essential for its use as a component in a larger ASN is that it is capable of operating effectively in environments with random payoff response characteristics. If for each context the output of the adaptive element only determines a probability for the payoff value, the adaptive element is capable of acting so as to increase its expected payoff value. It is beyond the scope of the present paper to thoroughly discuss these aspects of the adaptive element's behavior. The relevant theory is that of stochastic automation learning algorithms, and the reader is referred to the review by Narendra and Thathachar, (1974).

The Problem of Context Transitions

According to (2), the adaptive element uses the change in the payoff signal z as a factor determining weight

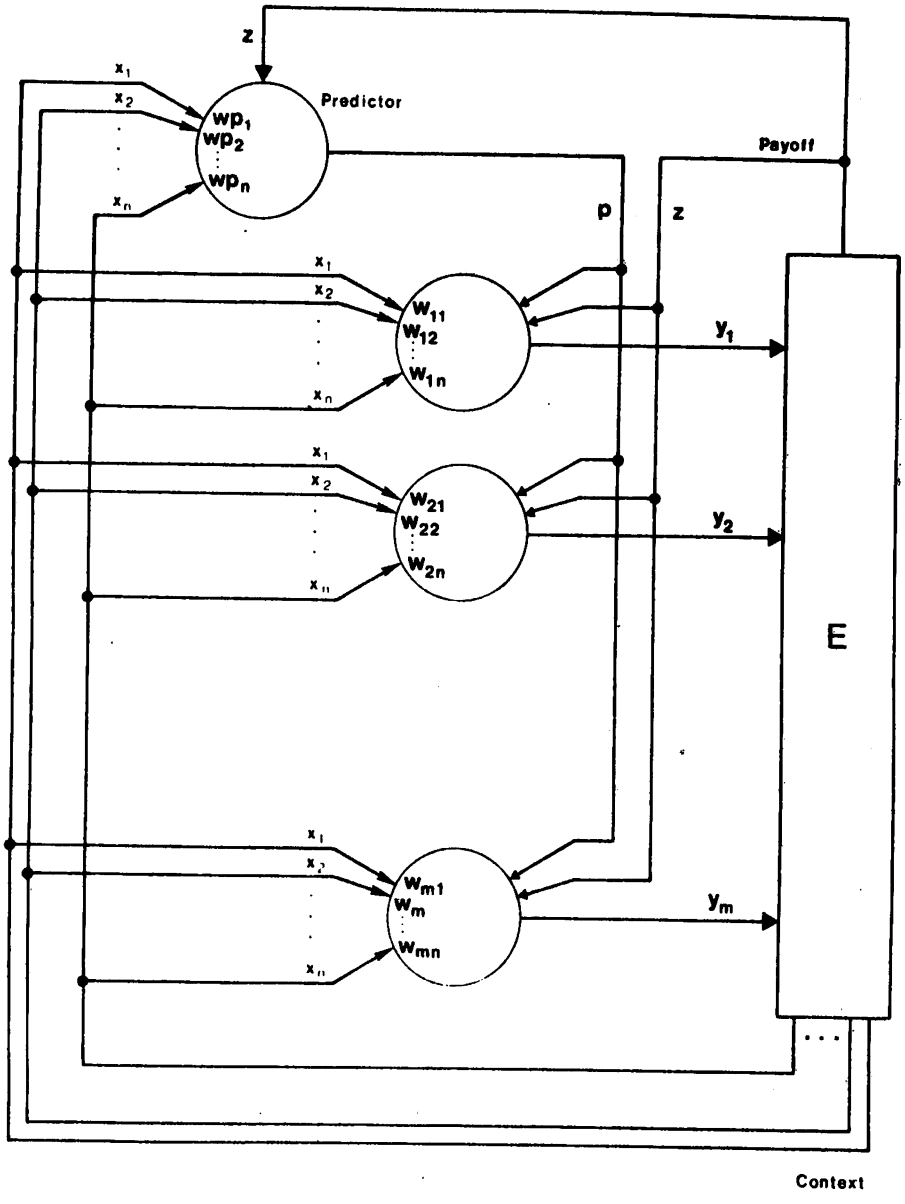

Fig. 3. An ASN consisting of m adaptive elements and one predictor. The adaptive element weights form an $m \times n$ associative matrix

solve the associative search problem. For each i, $i=1,\ldots,m$, let $\mathscr{X}_i^0=\{\mathbf{X}^x\in\mathbf{X}|y_i^x=0\}$ and $\mathscr{X}_i^1=\{\mathbf{X}^x\in\mathbf{X}|y_i^x=1\}$. That is, $\mathscr{X}_i^0$ ($\mathscr{X}_i^1$) is the set of all context vectors in which it is optimal for element i to produce output 0 (1). The sets $\mathscr{X}_i^0$ and $\mathscr{X}_i^1$ are linearly separable if there exists a real vector $\mathbf{W}_i=(w_{i1},\ldots,w_{in})$ such that

$$\mathbf{W}_i\cdot\mathbf{X}<0 \quad \text{if} \quad \mathbf{X}\in\mathscr{X}_i^0$$

$$\mathbf{W}_i\cdot\mathbf{X}>0 \quad \text{of} \quad \mathbf{X}\in\mathscr{X}_i^1.$$

We conjecture that for any n, $m>0$, there exist ASN parameters (c, cp, and the variance of the random variables) such that it can solve the associative search problem with as high a probability as desired if 1) each $\mathscr{X}^i$ is unimodal (i.e., does not possess suboptimal "peaks") and 2) $\mathscr{X}_i^0$ and $\mathscr{X}_i^1$ are linearly separable for each $i=1,\ldots,m$. The performance of learning automata in optimizing multimodal functions is a topic of current research.

Once this task is solved, the ASN functions as an associative memory similar to those discussed in the literature. For example, if a degraded context vector is presented, then the ASN can still perform an appropriate action if the degraded context vector is still sufficiently distinctive. Similarly, the ASN will produce actions in situations never before encountered by acting in a way appropriate in similar situations which it has experienced in the past. The ASN also exhibits the same resistance to damage shown by distributed associative memories (see Wood, 1978). In addition it is possible to prime the associative matrix with information likely to be useful for specific problem domains.

We note that if our conjecture is correct, perfect ASN performance does not require orthogonal context vectors. Associative memories have been discussed by Amari (1977) and Kohonen and Oja (1976) which are able to exhibit perfect recall if the keys are linearly independent but not orthogonal. Amari (1977) calls this orthogonal learning since it requires the orthogonalization of the set of keys. It can be shown that if the context vectors $\mathbf{X}^1, \ldots, \mathbf{X}^k$ are linearly independent, then $\mathscr{X}_i^0$ and $\mathscr{X}_i^1$ are linearly separable for each $i=1, \ldots, m$. This implies that if our conjecture is true, the ASN can solve the associative search problem if each Z^α is unimodal and the context vectors are linearly independent. This is an instance of orthogonal learning, but, as discussed above, it differs in that the ASN does not require the desired response for each key to be explicitly provided.

Examples

For illustrative purposes we let each payoff function Z^α in the following examples be a simple linear function of the ASN actions. To each context vector $\mathbf{X}^\alpha$ is associated a vector $\mathbf{Y}^\alpha=(y_1^\alpha, \ldots, y_m^\alpha)$ where $y_i^\alpha \in \{-1, 1\}$. We define Z^α as

$$Z^\alpha(\mathbf{Y}) = \mathbf{Y} \cdot \mathbf{Y}^\alpha$$

so that Z^α is maximized when each adaptive element i, $i=1,\ldots,m$, is "on" if $y_i^\alpha = +1$ or "off" if $y_i^\alpha = -1$. That is, Z^α is maximized by $\mathbf{Y}=(\mathbf{Y}^\alpha+1)/2$. We use the symbol $\mathbf{Y}^\alpha$ to denote both the 1, -1 valued vector $\mathbf{Y}^\alpha$ and the binary vector $(\mathbf{Y}^\alpha+1)/2$ since no confusion is likely to arise. Computing Z^α in this manner implies that if an adaptive element "turns on" in a context in which it should be on, of if it "turns off" in a context in which it should be off, then the value of Z^α will increase by 1 (assuming the other elements do not change their actions). Similarly, "turning on" when off is best or "turning off" when on is best decreases Z^α by 1. We do not claim that the optimization of such a simple linear function is a difficult task. Our intent here is to illustrate that a search is in fact performed by the ASN. More research is required to delineate the search capabilities of the ASN and related structures. In each of the following examples, the adaptive element learning constant $c=0.03$ and the standard deviation of each random variable is 0.1. In the cases using the predictor, $cp=0.1$.

a

$$\mathbf{X}^1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{Y}^1 = \begin{bmatrix} -1 \\ 1 \\ 1 \\ 1 \\ -1 \\ 1 \\ 1 \\ 1 \\ -1 \end{bmatrix} \quad \mathbf{X}^2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad \mathbf{Y}^2 = \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \\ 1 \\ -1 \\ -1 \\ 1 \\ 1 \end{bmatrix}$$

b

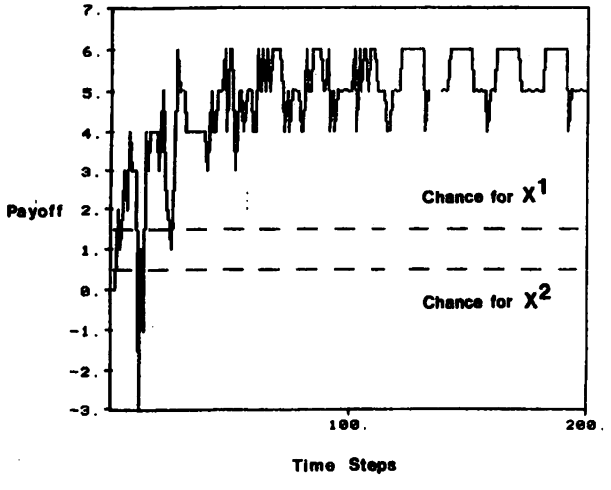

Fig. 4a and b. Example 1. **a** Two orthogonal context vectors $\mathbf{X}^1$ and $\mathbf{X}^2$ and the corresponding optimal output patterns $\mathbf{Y}^1$ and $\mathbf{Y}^2$. **b** Graph of payoff received by the ASN during a training sequence in which contexts were presented alternately, each held constant for 10 time steps. No predictor was used, but the learning constant c was set to zero for context transition. The dotted line represents the average payoff level obtainable if no learning occurred. The payoff received by the ASN increases over time and attains the optimal value for each context, i.e., 6 for $\mathbf{X}^1$, 5 for $\mathbf{X}^2$

Example 1. Figure 4 shows ASN behavior for the simplest case of two orthogonal context vectors $\mathbf{X}^1$ and $\mathbf{X}^2$ with $n=8$ and $m=9$. The optimal output patterns are determined by $\mathbf{Y}^1$ and $\mathbf{Y}^2$ (Fig. 4a). Notice that $Z^1(\mathbf{Y}^1)=6$ and $Z^2(\mathbf{Y}^2)=5$ so that a higher payoff is obtainable in context 1. The contexts were alternately presented, each held constant for 10 time steps. A predictor was not used. In order to prevent the transition from one context to another from providing misleading information, the learning constant c was momentarily set to zero while the context changed.

The dotted lines in Fig. 4b show the payoffs which could be expected in each context for output patterns generated purely by chance. The payoff actually received by the ASN increases over time and attains the optimal values for each context, i.e., 6 for context $\mathbf{X}^1$, 5 for context $\mathbf{X}^2$. After learning, the presentation of a context vector immediately "keys out" the pattern optimal for that context. Unlike other associative memory systems, however, the optimal patterns were never directly available to the system. Since the context patterns in this case have totally disjoint regions of non-zero values, the more interesting associative aspects of the system are not demonstrated. The resultant associative matrix simply stores the separate associations.

Figure 5 shows the behavior of the ASN for the same problem as illustrated in Fig. 4 with the exception that the learning constant c was not set to zero for context transitions. Learning occurs, but the almost perfect behavior shown in Fig. 4b is not attained even after 500 time steps. The reason for this is that the transition from $\mathbf{X}^1$ to $\mathbf{X}^2$ tends to penalize elements which may have been correctly responding to $\mathbf{X}^1$ since the payoff tends to decrease at the transition.

Figure 6 illustrates the behavior of the ASN with a predictor for the same problem shown in Figs. 4 and 5. The learning curve (Fig. 6a) is comparable to that obtained with c set to zero during transitions (Fig. 4b). Figure 6b shows the prediction error $p(t)-z(t+1)$ during the training sequence. The predictor comes to successfully predict that the highest payoffs in contexts $\mathbf{X}^1$ and $\mathbf{X}^2$ are respectively 6 and 5. Transitions from $\mathbf{X}^1$ to $\mathbf{X}^2$ do not penalize elements correctly responding to $\mathbf{X}^1$ since the payoff drop is "expected". Notice in Fig. 6 the errors committed approximately at time steps 400 and 450. Since we use normally distributed random variables to drive the search, there always remains a non-zero probability that an element will perform either action.

Example 2. Here $n=8$, $m=25$, and four non-orthogonal but linearly independent context vectors are considered (Fig. 7a). The optimal output patterns $\mathbf{Y}^1,\ldots,\mathbf{Y}^4$ are shown as 5×5 arrays, but should be

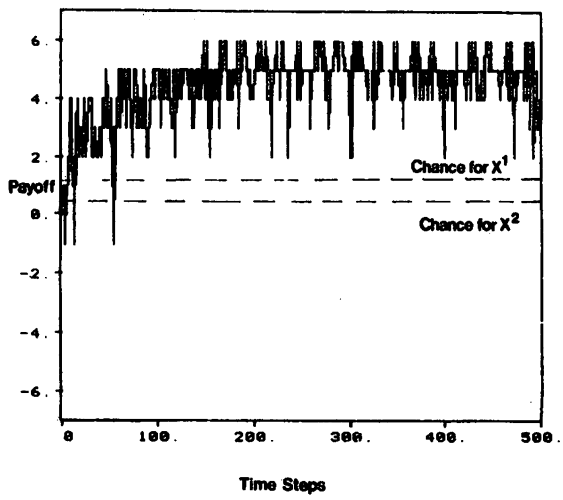

Fig. 5. The ASN payoff for the training sequence illustrated in Fig. 4 but with the learning constant held non-zero throughout. The perfect behavior shown in Fig. 4b is not attained

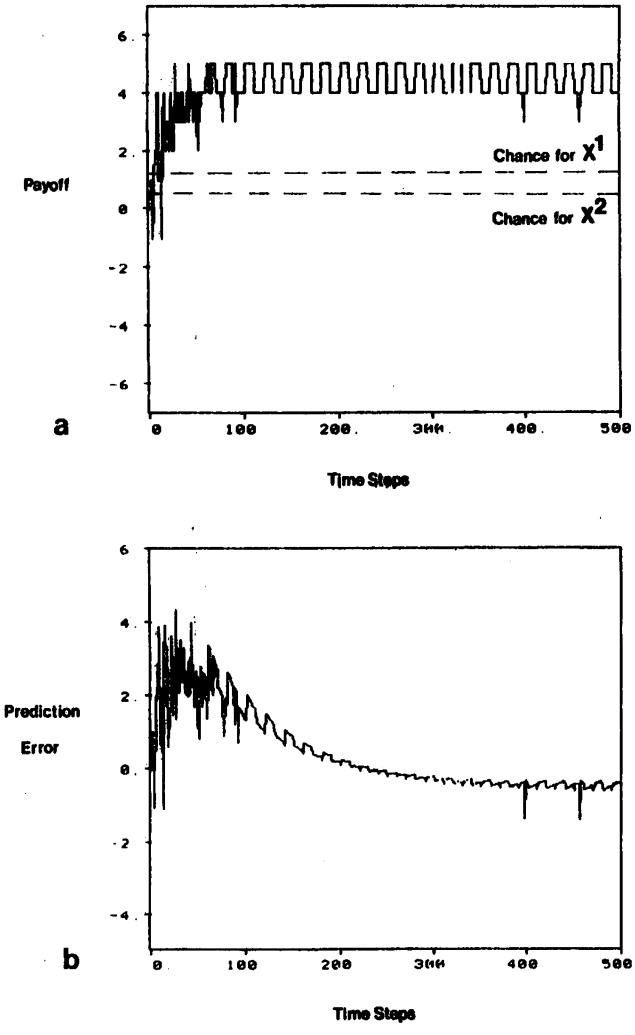

Fig. 6. a The ASN payoff for the training sequence illustrated in Fig. 4 but with the use of a predictor. b Prediction error $p(t)-z(t+1)$

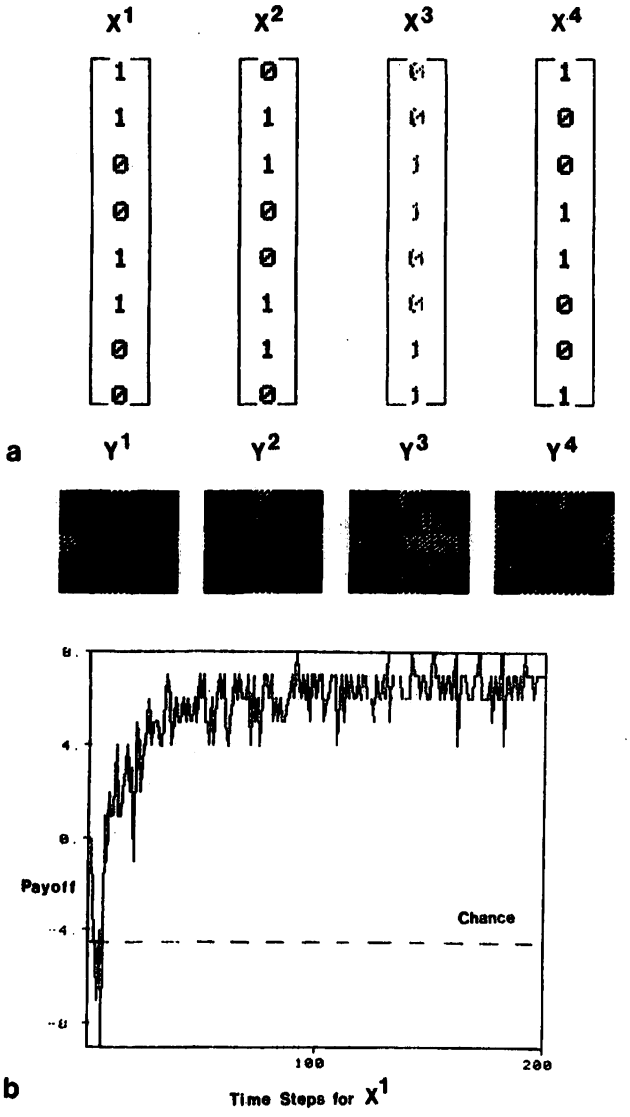

Fig. 7a and b. Example 2. **a** Four non-orthogonal but linearly independent context vectors and their corresponding optimal output patterns. **b** ASN payoff for time steps in which context vector $\mathbf{X}^1$ is present. There is a similar curve for each context vector

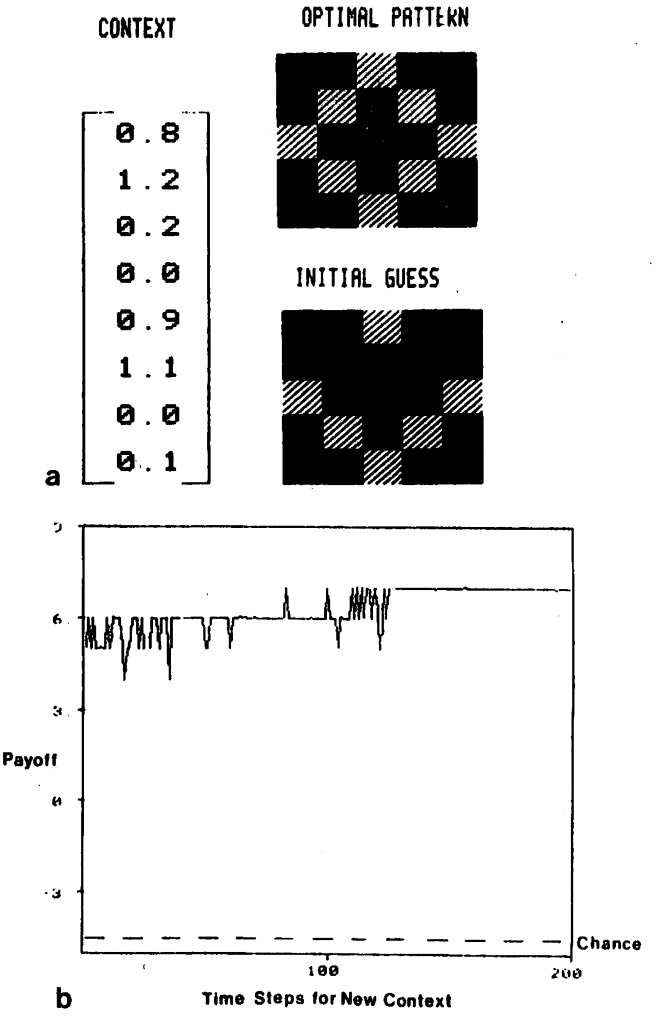

Fig. 8a and b. Example 3. With the associative matrix obtained after training in Example 2, context vector $\mathbf{X}^1$ of Fig. 7a was corrupted by additive noise and presented to the ASN. *E* implemented payoff function Z^1. **a** The corrupted context vector, the optimal output pattern, and the ASN's initial guess **b** ASN payoff as it searches for the optimal output pattern

thought of as "actions" and not as visual images. Again, each context was presented for 10 consecutive time steps, with the sequence repeating. No predictor was used. The learning constant was set to zero during context transition. After sufficient learning each context vector causes the retrieval of the optimal output pattern. This occurs even though the context vectors do not form an orthogonal set. Figure 7b shows the learning curve for context $\mathbf{X}^1$. The abscissa gives cumulative time steps in which context $\mathbf{X}^1$ was present. An ASN using a predictor has essentially the same behavior.

Example 3. With the associative matrix **W** containing the values obtained after training in Example 2, context vector $\mathbf{X}^1$ was corrupted by additive noise and presented to the ASN (Fig. 8a). As for other associative memories, keys corrupted by noise cause retrieval of patterns similar to the desired ones provided the corrupted key remains sufficiently distinguishable from the others. The pattern retrieved using the corrupted version of $\mathbf{X}^1$ resembles the stored pattern $\mathbf{Y}^1$. For the ASN, however, the retrieved pattern is just the *initial guess* (Fig. 8a) for the optimal pattern and the search resumes. Like most search procedures, the time

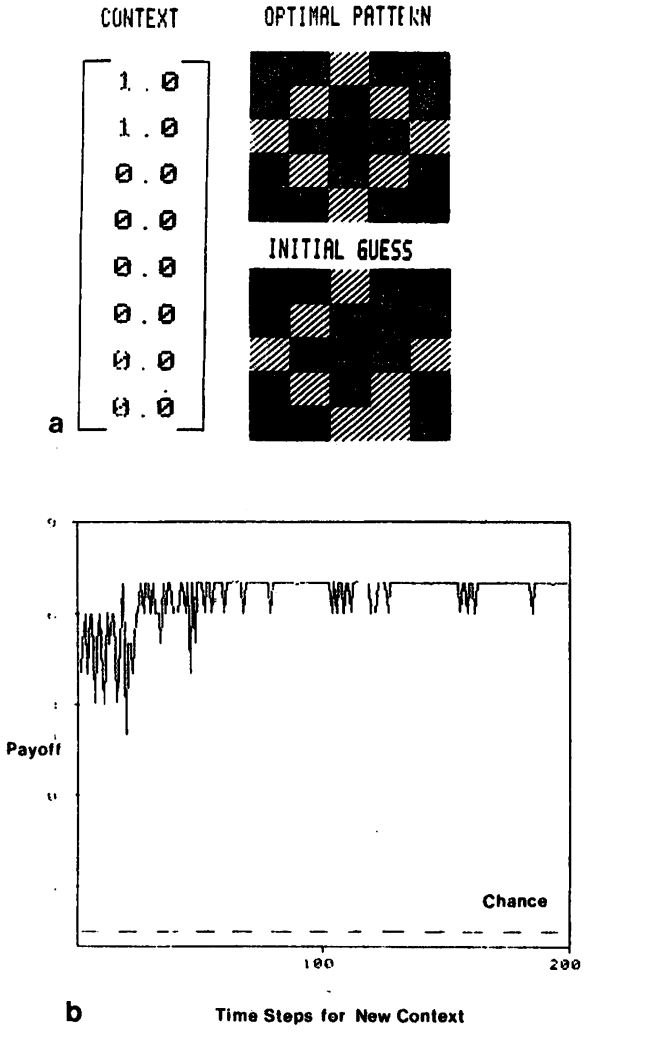

Fig. 9a and b. Example 4. With the associative matrix obtained after training in Example 2, a fragment of $\mathbf{X}^1$ was presented as context. *E* implemented payoff function Z^1. **a** The fragment of $\mathbf{X}^1$, the optimal pattern, and the initial guess. **b** ASN payoff as the search continues

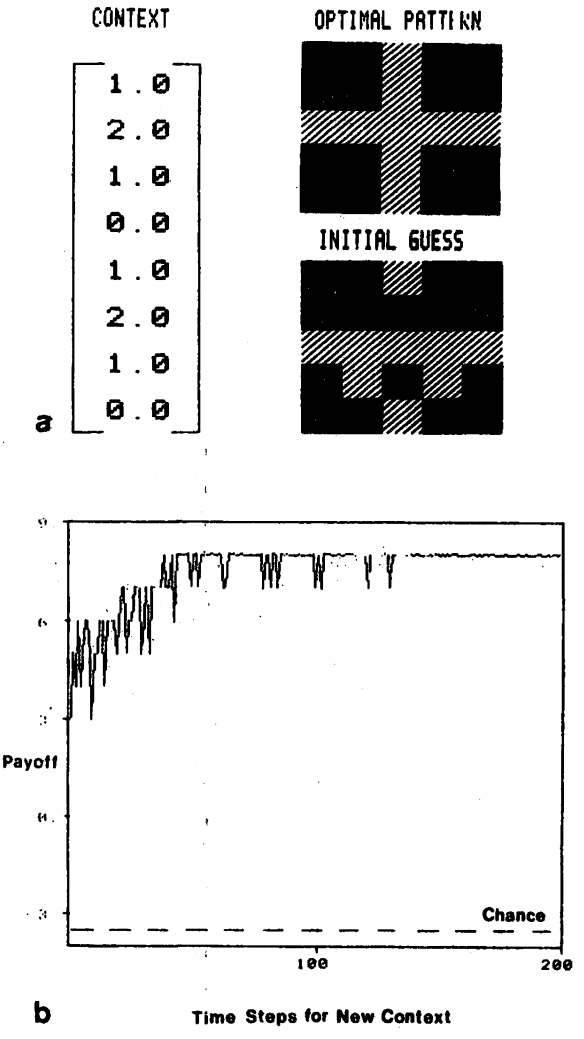

Fig. 10a and b. Example 5. The sum of $\mathbf{X}^1$ and $\mathbf{X}^2$ of Fig. 7a was presented as context to the ASN with the associative matrix obtained after training in Example 2. *E* implemented Z^2. **a** The context vector $\mathbf{X}^1+\mathbf{X}^2$, the optimal output pattern $\mathbf{Y}^2$, and the ASN's initial guess. **b** ASN payoff as the search continues

to convergence for the ASN is reduced if the initial guess is close to the optimal pattern. Hence, with the corrupted $\mathbf{X}^1$ being presented to the ASN and $\mathbf{Y}^1$ still the best output pattern, the ASN quickly corrects its response (Fig. 8b). At the conclusion of the search, the corrupted version of $\mathbf{X}^1$ is able to cause the immediate retrieval of $\mathbf{Y}^1$.

Example 4. Again with the associative matrix containing the values obtained by training in the four contexts of Example 2, a fragment of $\mathbf{X}^1$ is presented as a context vector (Fig. 9a). The pattern retrieved again acts as an initial guess and the ASN corrects it under control of environmental feedback (Fig. 9b).

Example 5. Here the sum of the two context signals $\mathbf{X}^1$ and $\mathbf{X}^2$ of Fig. 7a is presented as a context vector to the ASN, but the payoff function is the one previously signalled by $\mathbf{X}^2$ (that is, $\mathbf{Y}^2$ is best). In this case, the initial guess is a combination of the patterns $\mathbf{Y}^1$ and $\mathbf{Y}^2$ (Fig. 10a). Again the search process brings the initial guess to the optimal pattern (Fig. 10b).

Neural Search

The ASN arose from our investigation of the neural hypothesis of Klopf (1972, 1979, 1981). He hypothesized that neurons try to maximize their level of membrane depolarization by changing synaptic effectiveness in the following way: Whenever a neuron fires, those synapses that were active during the summation of potentials leading to the discharge become eligible to undergo changes in their transmission effectiveness. If the discharge is followed by further depolarization,

then the eligible excitatory synapses become more excitatory. If the discharge is followed by hyperpolarization, then eligible inhibitory synapses become more inhibitory. In this way a neuron will become more likely to fire in a situation in which firing is followed by further depolarization and less likely to fire in a situation in which firing leads to hyperpolarization.

The basic adaptive element operating according to (1) and (2) is very similar to Klopf's model of a neuron. The term $x_i(t-1)$ in (2) corresponds to Klopf's eligibility. A weight can change at time t only if there was activity on its pathway at $t-1$, i.e., $x_i(t-1) \neq 0$. More general forms of eligibility can be implemented by replacing this term with a more prolonged trace of activity as is discussed by Sutton and Barto (1981). The restricted form of eligibility used here is suitable because E always evaluates an output pattern in a single time step. The idea of eligibility is essential for the search behavior of an adaptive element since it permits the *consequences* of actions to influence the probability of these actions in the future. This cannot be accomplished by a Hebbian-type rule which associates simultaneous signals or nearly simultaneous signals with no sensitivity to which occurred earliest.

Unlike Klopf's hypothesized neuron, the adaptive element presented here tends to maximize a specialized payoff or reinforcement signal (z) rather than what would correspond to membrane potential (s). There are several interesting consequences of a rule that tends to maximize s. It permits secondary reinforcement to occur whereby the occurrence of a previously rewarded context itself is rewarding, and it may permit a single adaptive element to perform both the search and prediction tasks, eliminating the need for a separate predictor element. In this report we have focused only on the simpler case in which there is a specialized payoff or reinforcement signal.

The adaptive element presented here is an illustrative example of a class of adaptive mechanisms, some of which are more closely related to Klopf's hypothesis, and should not be literally interpreted as a model of a single neuron. In fact, we have purposefully referred to it as an adaptive element rather than a neural model. We do wish to suggest, however, that the general form of stochastic, closed-loop, reinforcement learning realized by the adaptive element merits close experimental investigation. Theory has shown that stochastic search procedures can be very effective means for the optimization of functions about which little is known. This capability combined with pattern recognition capabilities leads to considerable adaptive power. As a neural hypothesis, the adaptive element suggests that the stochastic component of neural discharge might perform the function of stochastic search. A closely related adaptive element is discussed with respect to behavioral and neurophysiological data in Sutton and Barto (1981).

Sensory-Motor Control Surfaces

It has been suggested that associative memories might provide effective means for the storage of sensory-motor associations required for sensory guided motor behavior (Albus, 1979). However, in every case there is the requirement for a signal to be present giving the "desired response" in order to form the correct sensory-motor association. Yet this kind of information is usually not available to an organism nor easy to obtain. After considerable experience in a given set of sensory contexts, the "desired response" for each context might become known through a learning process. But the associative memory structures proposed in the literature are not able to perform this type of learning. Their structure suggests how associations might be stored but does not address the very important questions concerning what information is chosen for storage. The ASN suggests how such questions might be explored.

Sensory-motor learning tasks provide natural examples of the type of problem the ASN is capable of solving. Sensory context is provided by exteroceptive and interoceptive stimulus patterns, and output patterns provide control signals to motor systems. Global reinforcement systems might provide information analogous to the ASN payoff signal. The associative matrix formed would implement a sensory-motor control surface. This interpretation of the ASN task suggests that research should continue in order to extend the ASN's capabilities in several different ways. 1) Most complex control tasks require nonlinear control surfaces. Elaboration of the ASN to permit the formation of nonlinear associations can be accomplished in the same manner as suggested for other associative memories in the literature (Poggio, 1975). 2) Most sensory-motor tasks have the property that the context which occurs next is partially a function of the control system's action. In the problem discussed in this report the ASN has no control over which context occurs. An interesting generalization of the ASN task is to require the ASN to control not only the payoff signal but also the context vectors in order to reach a context in which the highest payoff is available. This is a more general learning control problem. 3) The ASN task presented here is simplified by the occurrence of a payoff signal at every time step. In actual sensory-motor learning tasks the reinforcing events occur only occasionally. Secondary reinforcement capabilities would provide a first step toward the solution of this substantially more difficult problem.

Conclusion

The distributed memory properties of associative memory systems make them particularly interesting learning systems from both biological and theoretical perspectives. Although all associative memory systems described in the literature require the desired response for each key to be provided by some other source, the interesting properties of associative memory systems are not restricted to this form of learning. A more difficult type of learning, which can occur even if no part of the system or of the environment knows the desired behavior, is reinforcement learning. In this form of learning the environment provides only a performance measure of responses rather than desired responses, making the problem both more difficult for the learning system and less demanding for the environment. The ASN is an associative memory system capable of solving reinforcement learning tasks. Our results illustrate that the important properties of associative memories can be retained by a system capable of this more general and more difficult form of learning.

Acknowledgements. This research was supported by the Air Force Office of Scientific Research and the Avionics Laboratory (Air Force Wright Aeronautical Laboratories) through Contract No. F33615-77-C-1191.

References

Albus, J.S.: Mechanisms of planning and problem solving in the brain. Math. Biosci. **45**, 247–293 (1979)

Amari, S.: Neural theory of association and concept-formation. Biol. Cybern. **27**, 175–185 (1977)

Anderson, J.A., Silverstein, J.W., Ritz, S.A., Jones, R.S.: Distinctive features, categorical perception, and probability learning. Some applications of a neural model. Psychol. Rev. **85**, 413–451 (1977)

Cooper, L.N.: A possible organization of animal memory and learning. In: Proceedings of the Nobel Symposium on Collective Properties of Physical Systems. Lundquist, B., Lundquist, S. (eds.). New York: Academic Press 1974

Didday, R.L.: A model of visuomotor mechanisms in the frog optic tectum. Math. Biosci. **30**, 169–180 (1976)

Duda, R.O., Hart, P.E.: Pattern classification and scene analysis. New York: Wiley 1973

Freeman, W.J.: Mass action in the nervous system. New York: Academic Press 1975

Grossberg, S.: Adaptive pattern classification and universal recoding. II. Feedback, expectation, olfaction, illusions. Biol. Cybern. **23**, 187–202 (1976)

Harth, E., Tzanakou, E.: ALOPEX: a stochastic method for determining visual receptive fields. Vision Res. **14**, 1475–1482 (1974)

John, E.R., Schwartz, E.L.: The neurophysiology of information processing and cognition. Annu. Rev. of Psychol. **29**, 1–29 (1978)

Kasyap, R.L., Blaydon, C.C., Fu, K.S.: Stochastic approximation. In: Adaptive, learning, and pattern recognition systems: theory and applications, pp. 339–354. Mendel, J.M., Fu, K.S. (eds.). New York: Academic Press 1970

Klopf, A.H.: Brain function and adaptive systems – a heterostatic theory. Air Force Cambridge Research Laboratories research report AFCRL-72-0164, Bedford, MA. (1972) (AD742259). (A summary in: Proceedings of the International Conference on Systems, Man and Cybernetics, IEEE Systems, Man and Cybernetics Society, Dallas, Texas, 1974)

Klopf, A.H.: Goal-seeking systems from goal-seeking components: implications for AI. The Cognition and Brain Theory Newsletter, Vol. III, No. 2 (1979)

Klopf, A.H.: The hedonistic neuron: A theory of memory, learning, and intelligence. Washington, D.C.: Hemisphere 1981 (to be published)

Kohonen, T.: Associative memory: a system theoretic approach. Berlin, Heidelberg, New York: Springer 1977

Kohonen, T., Oja, E.: Fast adaptive formation of orthogonalizing filters and associative memory in recurrent networks of neuron-like elements. Biol. Cybern. **21**, 85–95 (1976)

Mendel, J.M., McLaren, R.W.: Reinforcement-learning control and pattern recognition systems. In: Adaptive, learning, and pattern recognition systems: theory and applications, pp. 287–317. Mendel, J.M., Fu, K.S. (eds.). New York: Academic Press 1970

Minsky, M.L., Papert, S.: Perceptron: an introduction to computational geometry. Cambridge, MA: MIT Press 1969

Nakano, K.: Associatron – a model of associative memory. IEEE Trans. Syst. Man Cybern. **3**, 380–388 (1972)

Narendra, K.S., Thathachar, M.A.L.: Learning automata – a survey. IEEE Trans. Syst. Man Cybern. **4**, 323–334 (1974)

Nilsson, N.J.: Learning machines. New York: McGraw-Hill 1965

Poggio, T.: On optimal nonlinear associative recall. Biol. Cybern. **19**, 201–209 (1975)

Rosenblatt, F.: Principles of neurodynamics; perceptrons and the theory of brain mechanisms. Washington: Spartan Press 1962

Sutton, R.S., Barto, A.G.: Toward a modern theory of adaptive networks: expectation and prediction. Psychol. Rev. (in press) (1981)

Tsetlin, M.L.: Automaton theory and modeling of biological systems. New York: Academic Press 1973

Widrow, B., Gupta, N.K., Maitra, S.: Punish/reward: learning with a critic in adaptive threshold systems. IEEE Trans. Syst. Man Cybern. **5**, 455–465 (1973)

Wigström, H.: A neuron model with learning capability and its relation to mechanisms of association. Kybernetik **12**, 204–215 (1973)

Willshaw, D.J., Buneman, O.P., Longuet-Higgins, H.S.: Non-holographic associative memory. Nature **222**, 960–962 (1969)

Wood, C.C.: Variations on a theme by Lashley: lesion experiments on the neural model of Anderson, Silverstein, Ritz, and Jones. Psychol. Rev. **85**, 582–591 (1978)

Received: December 1, 1980

Andrew G. Barto
Department of Computer and Information Science
University of Massachusetts
Amherst, MA 01003
USA