Biol. Cybern. 43, 175–185 (1982)

Biological Cybernetics

Synthesis of Nonlinear Control Surfaces by a Layered Associative Search Network

Andrew G. Barto, Charles W. Anderson, and Richard S. Sutton
Department of Computer and Information Science, University of Massachusetts at Amherst, Amherst, USA

Abstract. An approach to solving nonlinear control problems is illustrated by means of a layered associative network composed of adaptive elements capable of reinforcement learning. The first layer adaptively develops a representation in terms of which the second layer can solve the problem linearly. The adaptive elements comprising the network employ a novel type of learning rule whose properties, we argue, are essential to the adaptive behavior of the layered network. The behavior of the network is illustrated by means of a spatial learning problem that requires the formation of nonlinear associations. We argue that this approach to nonlinearity can be extended to a large class of nonlinear control problems.

1. Introduction

Nonlinearity is an important property of most pattern recognition and control tasks, and the inability of learning systems composed of neuron-like elements to handle nonlinearity in an extensible way has formed the basis of many criticisms of this approach to problem solving and its relevance to biological information processing (e.g., Minsky, 1961; Minsky and Papert, 1969). In this article we present an associative memory network composed of neuron-like adaptive elements that is capable of solving a class of nonlinear control problems. The network is an extension of the associative search network (ASN) described previously by Barto et al. (1981) that employs a novel type of adaptive element based on the theory of Klopf (1972, 1979, 1982). The control problem with which we illustrate its behavior is an extension of the landmark learning task presented by Barto and Sutton (1981a). In this type of problem, the ASN controls movement in a spatial environment and forms associations between optimal directions of movement and stimulus patterns determined by its position with respect to a configuration of landmarks. While suggestive of animal learning behavior, these illustrations are not intended to be realistic models of the behavior of any particular animal. Barto and Sutton (1981a) point out that the spatial environment can be interpreted as a more abstract type of space, such as the state space of a dynamical system.

2. Approaches to Nonlinearity

Nonlinearity is not really a property of a problem per se, but rather a property of a particular way of representing a problem in terms of a set of variables, usually called features, properties, or predicates (see Minsky and Papert, 1969). A problem is linear for a given representation if the desired outputs of the pattern recognizer or controller are linear functions of the representation variables. A variety of well-known algorithms exist, and can be implemented by networks of neuron-like elements, that are able to find the correct weighting factors for the contribution of each representation variable to each output function of the adaptive system (e.g., Amari, 1977; Duda and Hart, 1975; Sutton and Barto, 1981). Many approaches to solving problems that are not linear in terms of a given representation involve specifying higher dimensional representations in which the problem is linear. For example, a problem that is not linear in terms of the variables x and y may be linear in terms of the variables x, y, x^2, y^2, and xy, and coefficients can be found by existing linear learning rules to express a desired function as a weighted sum of these five variables. This is the approach discussed by Poggio (1975).

The additional feature variables need not be products of the original variables but can be arbitrary functions of these variables as discussed, for example, by Nilsson (1965) and Minsky and Papert (1969). A

0340-1200/82/0043/0175/$02.20

straightforward instance of this approach is provided by a method that explicitly divides the feature space into a large number of small regions so that a different system action can be associated with each region. For example, the BOXES system of Michie and Chambers (1968) uses a representation in which the control space is divided into 225 independently accessable "boxes", and a similar scheme is used in the sensorimotor learning system of Raibert (1978). Albus (1979) proposes a related coding scheme in which the regions are not disjoint. These table-lookup approaches are memory intensive and require a priori selection of a sufficiently fine representation. Moreover, a representation that is too fine results in poor generalization capabilities and needlessly slow learning.

Various methods have been proposed for adaptively generating features rather than requiring them to be specified a priori. The central ideas in most of these methods are to generate features that are "like" previously useful features or to form nonlinear combinations of features that have proven useful. Minsky (1961) discusses this problem and examples are provided by Klopf and Gose (1969), Selfridge (1955), and Ivakhnenko's method of groups (1971). Michie and Chambers suggest that their BOXES system could be improved by the addition of mechanisms for "splitting" boxes when finer discriminations are needed and for "lumping" boxes that are associated with the same control action. They do not, however, provide a mechanism for doing this.

Although the system described here does not rely on a representation consisting of a large number of disjoint "boxes", we were motivated by Michie and Chamber's suggestion of splitting as a useful method of representation development. We use a network consisting of two layers. The output layer is a linear ASN as discussed by Barto et al. (1981) and is thus subject to all of the limitations of linearity including those emphasized by Minsky and Papert (1969). The input layer, however, is designed to adaptively form a representation in terms of which the problem can be solved linearly by splitting each of the input features.

3. The Linear Landmark Learning Problem

We briefly describe the linear landmark learning problem and the linear ASN capable of solving it (Fig. 1) that was presented by Barto and Sutton (1981a)[1] and then extend the problem and the network to the nonlinear case. Figure 1A shows a spatial environment consisting of a central landmark (shown as a tree) surrounded by four other landmarks (shown as boxes and circles). Thinking of this as an olfactory environment for a simple organism, each landmark emits a distinctive "odor" that decays with distance. The "odors" extend as far as the large circles shown in Fig. 1A. The "odor" of the central landmark will act as an attractant for the network. The asterisk shows the location of the ASN. The ASN's input pattern is therefore determined by its location in this environment.

Figure 1B shows an ASN with four input pathways labeled vertically according to the landmarks to which they respond. The lowermost "payoff" input is a specialized pathway responding to the attractant distribution produced by the tree. The four output pathways labeled horizontally at the bottom each produce a 0 or 1 at each time step and determine the direction of movement of the network. For example, if $N=0$, $S=1$, $E=1$, and $W=0$ (as shown by the shaded output elements in Fig. 1B), the network will move a fixed distance south and east. Connection weights between input and output elements are shown as circles centered on the intersections of the input pathways with the element "dendrites". Positive weights appear as hollow circles, and negative weights appear as shaded circles. Circle size codes weight magnitude.

The ASN's task in this environment is to 1) find the central tree landmark by climbing the attractant distribution and 2) associate with each sensory input pattern (and hence with each place in the environment) that action which causes movement toward the tree. These place-action associations are to be stored by means of the network's matrix of connection weights; they are never explicitly available in the environment. As a result of learning these place-action associations, the network can proceed directly to the tree by "reading out" the action associated with each position along its path, even in the absence of the attractant distribution. Since for the environment just described, the correct associative mapping is linear in terms of the stimulus patterns, the linear ASN shown in Fig. 1B is able to solve the landmark learning problem by forming the weights shown in Fig. 1C. The operation of this ASN is fully described by Barto and Sutton (1981a) and is identical to that of the second layer of the network described below. Figure 1D shows the results of learning in vived form as a vector field giving the expected direction of the network's movement through each position in space. This vector field is determined from the network's weight values and is never literally present in the environment.

In another experiment described by Barto and Sutton (1981a), the ASN was allowed to learn in the environment just described, and then the box shaped landmarks were interchanged. Figure 1E shows the

1 Our presentation here differs slightly from that of Barto and Sutton (1981a). The symbols for the landmarks and the ordering of sensory input pathways to the network are different

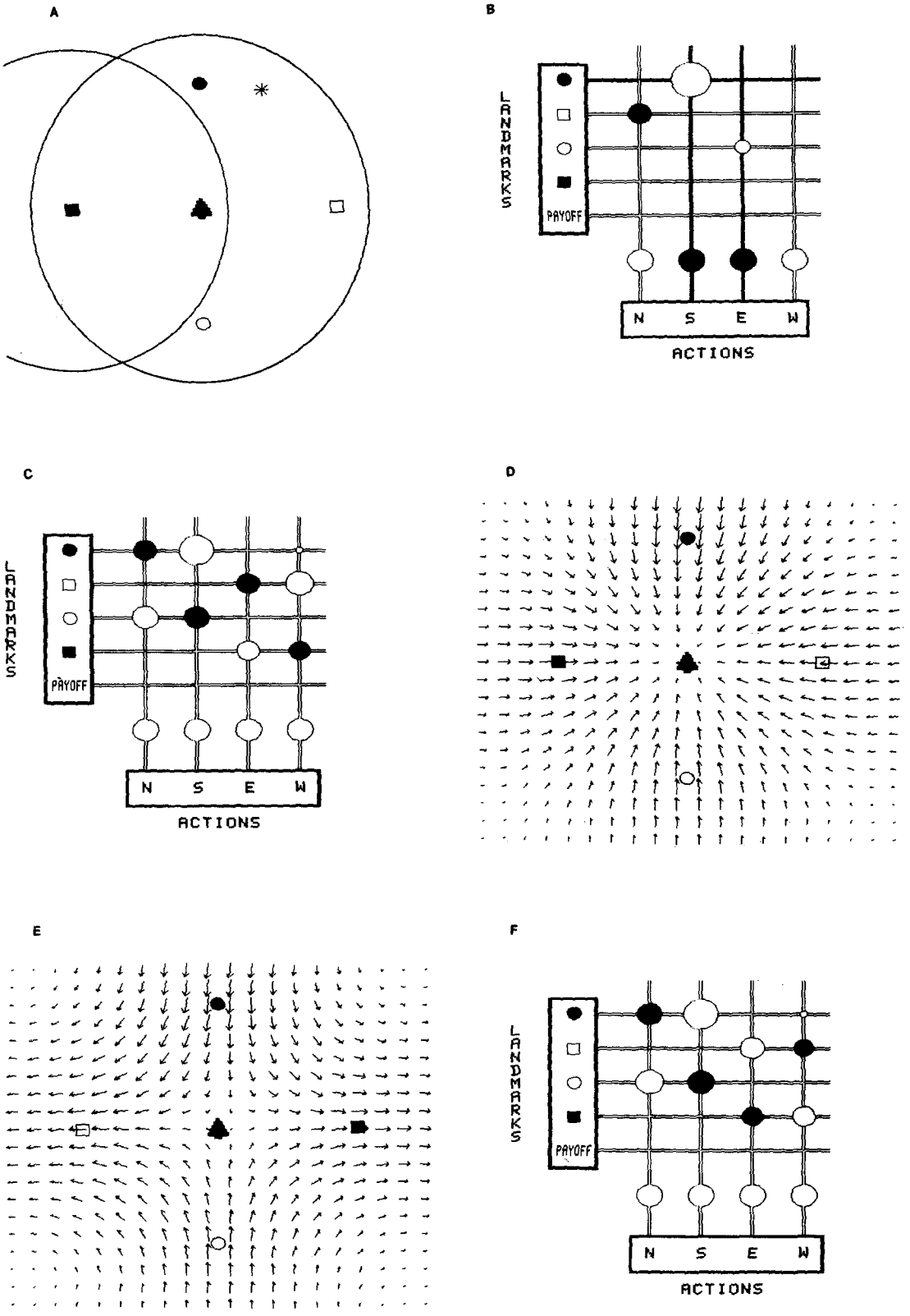

Fig. 1A–F. A *linear landmark learning* problem. **A** A spatial environment. **B** A linear associative search network for controlling locomotion. Positive weights appear as hollow circles; negative weights appear as shaded circles. **C** The configuration of the network after it has solved the problem. **D** A vector field representation of the contents of the network's memory after it has solved the problem. **E** The vector field showing how the network would tend to move if the locations of the box landmarks were interchanged after learning. **F** The network after relearning in the altered environment

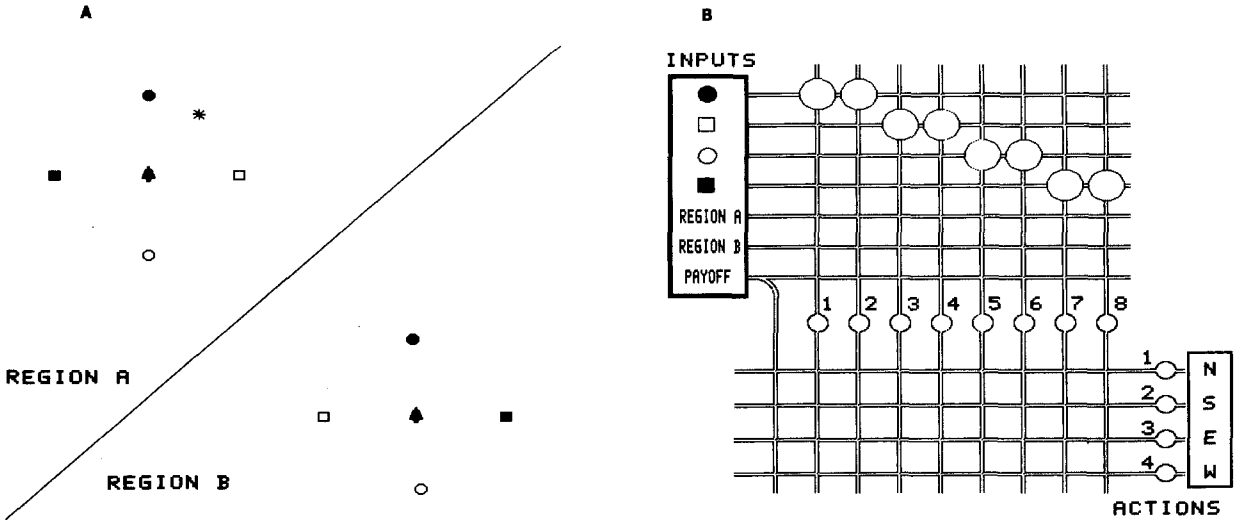

Fig. 2A and B. A nonlinear landmark learning problem. **A** An environment with two regions labeled "region A" and "region B". **B** A two-layer network. Layer 2 is identical to that shown in Fig. 1B except that it has eight input pathways in addition to the payoff pathway

vector field resulting from evaluating the ASN's associative matrix in the altered environment. The ASN is initially misled by its sensory information but quickly relearns to the altered environment resulting in the associative matrix shown in Fig. 1F. If we were to change the environment back to its original configuration, the ASN would change its associative matrix back to that shown in Fig. 1C. Thus, it is clear that the ASN as described is not capable of maintaining both control surfaces at the same time. As it learns in an environment with a different configuration of the same landmarks, it "re-writes" its memory, erasing traces of previous learning. This suggests the following task, which turns out to be nonlinear in terms of the landmark signals.

4. A Nonlinear Landmark Learning Problem

Figure 2A shows an environment containing two areas labeled "region A" and "region B". Corresponding landmarks produce the same sensory signals in both regions (e.g., the shaded box "smells" the same in both regions), but sensing a box landmark should produce movement in opposite directions in the two regions. That is, "hollow box in region A" should be associated with movement west, but "hollow box in region B" should be associated with movement east. Similarly, the correct associations for the shaded box depend on the region in which it is sensed. We consider the case in which there exist features, detectable by the network, that distinguish region A from region B. In the most general case, these distinguishing features may be complex patterns or relationships between more basic features, but for simplicity, and without undue loss of generality given our purposes, we simply assume that there is a sensor that is activated whenever the system is in region A and one that is activated whenever it is in region B. A signal from one of the region sensors must be capable of switching the effects of the two box landmarks on the east and west output elements in opposite senses. This cannot be accomplished by the sort of linear mapping the network shown in Fig. 1B is capable of forming (this is proved in detail in Appendix A).

What seems to be needed are signals distinguishing the sensing of a landmark in region A from sensing that same landmark in region B. Figure 2B shows a network consisting of two layers of adaptive elements (questions about why the network takes this particular form and why it is able to solve problems of this type will be discussed in a more general setting below). The output layer shown at the bottom, which we call layer 2, is identical to that shown in Fig. 1B except that it has eight rather than four input pathways in addition to the tree or payoff pathway. The input layer, which we call layer 1, consists of eight adaptive elements each receiving input from the four landmarks, the region A and region B indicators, and the tree.

The eight elements of layer 1 are organized in pairs: elements 1 and 2, elements 3 and 4, etc. The elements in each pair inhibit one another so that only the most strongly stimulated element of each pair can be active at any time. The large positive connection weights in layer 1 are all set permanently to the same value. Consequently, before any learning takes place in layer 1, the layer 1 elements simply transmit the layer 1 input signals to layer 2, sometimes via one element of

each pair and sometimes via the other (so that this network can also solve the linear problem described above). If the task cannot be solved linearly, then the paired elements will differentiate, or "split", in terms of the input patterns to which they are tuned and the influences they exert on layer 2 elements. The layer 2 elements are also paired so that at each time step only one element in each of the north/south and east/west pairs is active [if, however, ε in Eq. (3) below is nonzero, then both elements of each pair can be active with a probability depending on the size of ε]. This merely serves to keep the network moving efficiently and is not an important feature of the system.

Let $x_1(t), \ldots, x_6(t)$ denote the signals at time t from the landmarks and the region indicators in the order shown in Fig. 2B and let $z(t)$ denote the attractant signal from the tree. Let $y_1^1(t), \ldots, y_8^1(t)$ and $y_1^2(t), \ldots, y_4^2(t)$ respectively denote the outputs of the elements of layer 1 and layer 2. Finally, let $w_{ij}^1(t)$, $i=1, \ldots, 8$, $j=1, \ldots, 6$, denote the connection weight at time t between input pathway x_j and element i in layer 1, and let $w_{ij}^2(t)$, $i=1, \ldots, 4$, $j=1, \ldots, 8$, denote the connection weight at time t between element i in layer 1 and element j in layer 2. In order to represent the pairing of the layer 1 elements, let $\bar{i}$ be the element paired with element i, $i=1, \ldots, 8$. Thus, if $i=1$, then $\bar{i}=2$, etc. We denote the pairing of the north/south and east/west elements in layer 2 in the same manner. The layer 1 weights shown as large circles in Fig. 2B are fixed at the value 1.

The elements of layer 1 operate as follows. For each time $t=0, 1, \ldots$ and each $i=1, \ldots, 8$, let

$$s_i^1(t) = \sum_{j=1}^{6} w_{ij}^1(t) x_j(t) + \mathrm{NOISE}_i^1(t) \tag{1}$$

denote the weighted sum of the input signals to element i of layer 1, plus a random number $\mathrm{NOISE}_i^1(t)$ sampled from a mean zero normal distribution. Then the output of element i of layer 1 is

$$y_i^1(t) = \begin{cases} \max(0, s_i^1(t)) & \text{if } s_i^1(t) > s_{\bar{i}}^1(t) \\ 0 & \text{otherwise.} \end{cases}$$

This means that at each time step only one element of each pair has nonzero output. The active element is the one having the largest sum of input stimulation and a random number.

The layer 2 elements operate in a similar manner. For $i=1, \ldots, 4$, let

$$s_i^2(t) = \sum_{j=1}^{8} w_{ij}^2(t) y_j^1(t) + \mathrm{NOISE}_i^2(t) \tag{2}$$

and

$$y_i^2(t) = \begin{cases} 1 & \text{if } s_i^2(t) - s_{\bar{i}}^2(t) > \varepsilon \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Thus, the outputs of layer 1 elements act as inputs to layer 2 elements; and, whereas the outputs of layer 1 elements have positive real values, the outputs of layer 2 elements are binary valued.

The network interacts with the environment in such a way that the values of the input signals at any time t depend on the position of the network in the environment at time step $t-1$ together with the layer 2 element output values at time step $t-1$. The connection weights of each layer are updated based on the input received, the action taken, and its consequences in terms of a change in attractant level z. In particular, except for the fixed weights in layer 1, the connection weight values are determined through these difference equations:

$$w_{ij}^1(t) = w_{ij}^1(t-1) + c_1[z(t) - z(t-1)] \cdot [y_i^1(t-1) - y_i^1(t-2)] x_j(t-1), \tag{4}$$

$$w_{ij}^2(t) = w_{ij}^2(t-1) + c_2[z(t) - z(t-1)] \cdot [y_i^2(t-1)] y_j^1(t-1). \tag{5}$$

Equation (4) implies that the weight corresponding to the connection between input pathway j and layer 1 element i increases if an increase in element i's activity in the presence of input signal x_j is followed by an increase in attractant level z. Equation (5) implies that the weight corresponding to the connection from layer 1 element j to layer 2 element i increases if layer 2 element i "fired" in the presence of a signal from layer 1 element j, and this is followed by an increase in the attractant level z. The layer 1 and layer 2 connection weights change according to these slightly different rules because layer 1 elements have real valued activity whereas layer 2 elements are binary. See Barto et al. (1981) and Barto and Sutton (1981a, b) for additional discussion of this class of learning rules[2]. Appendix B contains detailed information regarding parameter values and protocols of the computer simulation experiment we describe next.

We first place the network in region A where it climbs the attractant distribution due to the presence of the tree and produces the trail shown in Fig. 3A. At the same time, it forms associations between its stimulus patterns and the optimal actions. These associations are shown in vector field form in Fig. 3B. Notice that the associations are correct for region A but are incorrect for region B. This is because the

2 The timing of the weight changes implied by Eqs. (4) and (5) differs slightly from that implied by the rules discussed in these references. A one time step delay between the calculation of a change in weights and the use of the weights in choosing an action is eliminated in the network presented here. This increases the rate of learning slightly but does not qualitatively change the behavior of these systems for any of the problems we have studied

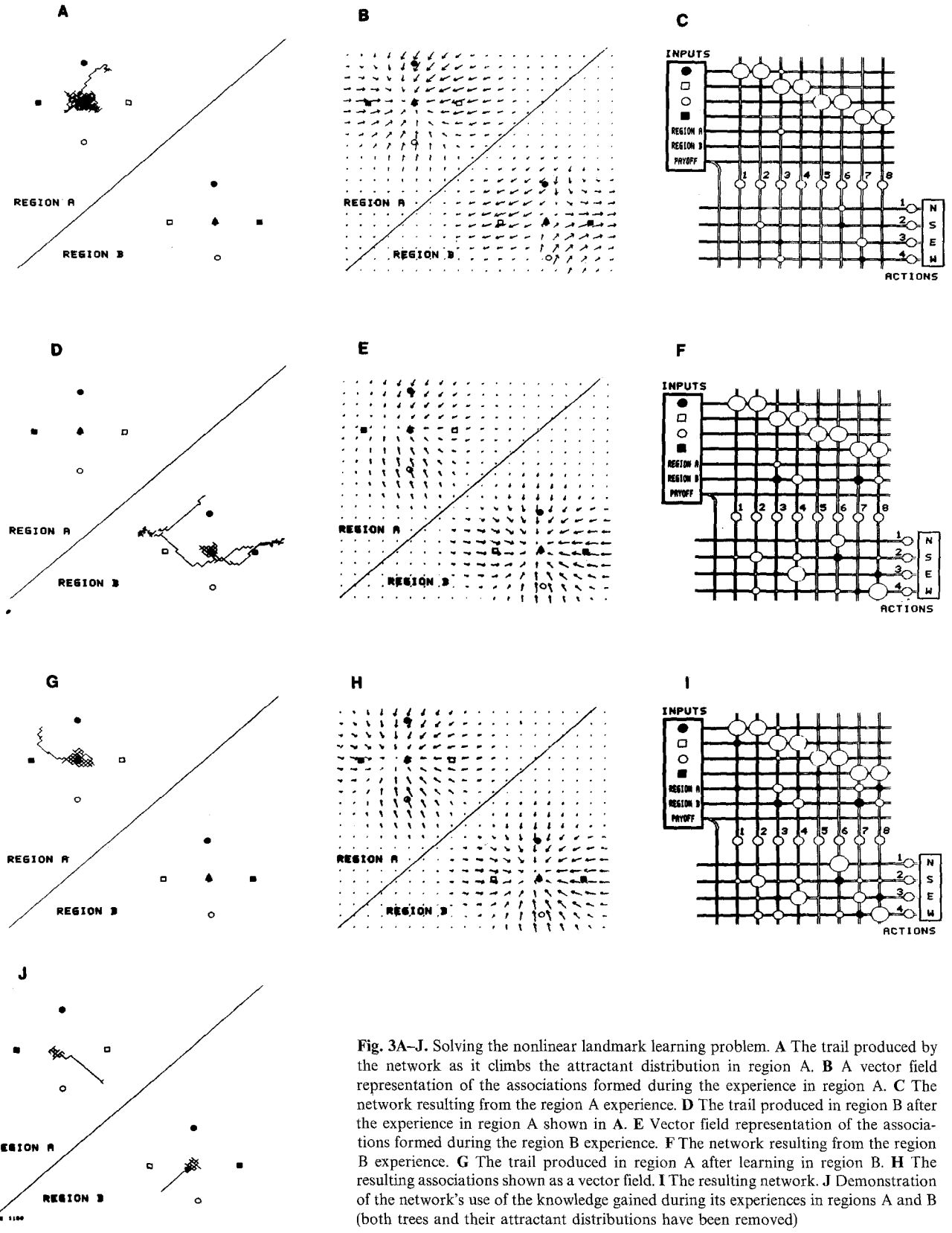

Fig. 3A–J. Solving the nonlinear landmark learning problem. **A** The trail produced by the network as it climbs the attractant distribution in region A. **B** A vector field representation of the associations formed during the experience in region A. **C** The network resulting from the region A experience. **D** The trail produced in region B after the experience in region A shown in **A**. **E** Vector field representation of the associations formed during the region B experience. **F** The network resulting from the region B experience. **G** The trail produced in region A after learning in region B. **H** The resulting associations shown as a vector field. **I** The resulting network. **J** Demonstration of the network's use of the knowledge gained during its experiences in regions A and B (both trees and their attractant distributions have been removed)

network's generalization from its experience in region A to region B is inappropriate due to the reversed box landmarks. The network resulting from this experience is shown in Fig. 3C. Notice that there has been a tendency for one element of each of the layer 1 element pairs to become tuned to respond strongly to various patterns of landmark "odors" in region A and less strongly to these patterns outside of region A (since positive weights form from the region A sensor to these elements). These elements initially happen to be active more frequently and, as they begin to be excited by the region A input pathway, their probabilities of activity steadily increase. During this period, the connections from these elements to layer 2 are established in a manner appropriate for moving in region A. Thus a control surface appropriate for directing action in region A is formed.

What happens when the organism is then placed in region B? Initially, the region A control surface is accessed since the layer 1 elements that are tuned to respond strongly to certain "odor" patterns in region A also respond to these patterns in region B, although less strongly. This results in the trail shown in Fig. 3D and can be seen as the network's attempt to generalize its region A experience to region B. Having been placed north of the shaded circular landmark, the network proceeds almost directly south and west as a result of being correctly directed by the shaded circle and incorrectly directed by the hollow box. These actions are punished since the network moves down the attractant gradient in region B. This tends to "erase" the region A control surface. However, this also causes inhibitory connections to form from the region B sensor to the elements selected in region A. This steadily decreases the probability that the elements selected in region A become active in region B. Then, whenever activation switches to the untuned element of a pair (and the probability of this steadily increases) *and* the network happens to move in the correct direction, then this element will begin to be tuned to respond to an "odor" pattern in region B and therefore provide a signal to layer 2 that can be associated with the correct actions for region B. Consequently, the erasure of the region A control information eventually stops as new associations are formed appropriate for region B (Fig. 3F). Continued exploration results in the formation of the associations shown in Fig. 3E as a vector field. New experience in region A quickly reinstates any lost information (Fig. 3G–I). By examining Fig. 3I, one can see that the layer 1 elements have tuned themselves to represent the environmental features as follows:

Element 1: unused
Element 2: shaded circle in both regions
Element 3: hollow box in region A
Element 4: hollow box in region B
Element 5: unused
Element 6: hollow circle in both regions
Element 7: shaded box in region A
Element 8: shaded box in region B.

Layer 2 can therefore generate the appropriate actions even though they are restricted to being linear functions of its input patterns. Although this process sounds complicated, it does in fact occur with great reliability and is not overly sensitive to parameter values.

Figure 3J shows the network behavior as the information is used that was stored during the experiences we have described. Both trees and their attractant distributions have been removed, and the network is started from places it has never before visited. Its path in each region shows direct approach to the former location of the tree.

5. Comments on the Problem and the Network

The particular landmark learning problem and network we have described are too simple to illustrate clearly some properties of the general approach we are suggesting. First, we have assumed the existence of single distinguishing features for each region and have provided input pathways to the network for these features. Although we labeled these pathways "region A" and "region B" and did not provide them with fixed connections in layer 1, they do not play specialized roles in the structure of the network. If it had been necessary for the problem's solution, the network could have formed features by combining any of its input variables. A two-layer network such as described here can solve these problems as long as the regions are distinguishable by linearly separable patterns of the problem's variables. This does not, however, imply that these problems as wholes can be solved linearly.

Another set of issues relates to the small size of the network. One might wonder, for example, why we did not just supply layer 2 from the start with all possible pairwise combinations of the landmark and region variables rather than requiring layer 1 to form the necessary combinations. There are two answers to this. First, the strategy of supplying all possible combinations of variables quickly leads to obvious combinatorial difficulties as problems become larger. The approach illustrated by our network is one in which enough structure (i.e., hardware) is provided for a fixed number of combinations, and the network itself must form the most useful combinations consistent with this structural constraint. A second reason for requiring the network itself to form combinations of variables only when necessary is that generalization capabilities are facilitated. Unnecessarily "splitting" a variable

prevents generalization from taking place along that dimension of the representation. This is illustrated by the example shown in Fig. 3. After experience in region A, the network attempted to use the relationships learned in region A to guide its behavior in region B. The relationships involving the boxes happened to be inappropriate in region B, but those involving the circles successfully generalized. The network immediately moved south when placed in the northern part of region B (Fig. 3D). If separate variables for each landmark in each region had been initially supplied to the layer 2 network, then no use of the region A experience would have been attempted in region B, and learning would have been slower.

6. Lateral Inhibition and the Enforcement of Variety

The network we have described does not explicitly employ a neural-like mechanism for restricting activity to one element of each pair (we simply select the element having maximal excitation). A variety of modeling efforts have shown that a lateral inhibitory network of neuron-like elements with the appropriate dynamics can select the maximally activated element and suppress the activity of all others. The reticular formation model of Kilmer et al. (1969) was the first to employ this process, and many other related models have been described (e.g., Amari and Arbib, 1977; Didday, 1976; Fukushima, 1973, 1980; Grossberg, 1976a, b; Spinelli, 1970; von der Malsburg, 1973). It should be clear that an explicit lateral inhibitory structure could be added to our network in order to perform the selection process, and we view the pairs of layer 1 elements as the simplest instances of populations of mutually inhibiting elements. Our network employs the selection process for the same reason that others have put forward for lateral inhibition, namely, to enforce variety during a learning process. For example, von der Malsburg's (1973) model of the development of orientation columns in visual cortex posits lateral inhibition in order to prevent neighboring cells from becoming tuned to the same optimal orientations. If no selection process were employed in our network, then there would be no tendency for the elements of each pair to become tuned to different input patterns and to cause different effects on layer 2. This differential tuning might still occur, but with a much lower probability.

7. Search and Layered Networks

We have called networks of the type described above associative search networks because they autonomously generate activity patterns via a random process that becomes biased as learning proceeds toward producing patterns of higher payoff. The elements of which these networks are composed differ from those previously studied in two fundamental ways. First, the random component of each element's activity is essential to the learning process since it generates trials in the absence of any pre-established influence from sensory input. Second, the feedback from the environment that governs the learning process is a reinforcement signal rather than a signal that provides either the difference between the pattern generated and the optimal pattern (a signed-error) or the optimal pattern itself. This latter type of signal is employed by most adaptive elements previously studied, including those forming various associative memory structures that use Hebbian or perceptron learning rules. Either type of these previously studied systems can only learn if its environment actually knows what the optimal responses are for a training sequence of stimulus patterns. A reinforcement learning system, on the other hand, can learn to produce optimal patterns in environments that can only provide evaluations of the system's actions. A rather subtle but important point here is that an adaptive system's environment can evaluate the system's actions, that is, can provide payoff or reinforcement signals, without containing explicit knowledge of what would constitute an optimal action. See Barto and Sutton (1981b) for a more complete discussion of these and related issues.

These characteristics of the learning rules we employ permit the layered network architecture to function effectively. We might expect the layered network's environment to know the optimal actions of the elements in the network's output layer for a training sequence, but it is generally impossible for the environment to know, even for a training sequence, how each element in the preceding layers should behave for each situation. If the interior elements of the network (i.e., those elements that are not output elements) can only adjust their parameters appropriately if they are explicitly instructed how to respond, then one would not expect useful behavior to result since most environments simply cannot provide such detailed information. On the other hand, elements in the interior of a layered network that are capable of reinforcement learning can suitably adjust their parameters on the basis of the environment's evaluations of the network's overall performance. If an action of an interior element is followed by improved network performance, then the occurrence of that action is made more likely. The environment need not know what the action was nor what it should have been[3]. Despite considerable effort,

3 Rosenblatt (1962) considered this problem but proposed a different method for solving it. He proposed schemes to compute local error signals rather than to use environmental evaluations

little success has been achieved in designing layered adaptive networks capable of solving nonlinear pattern recognition and control problems[4]. We believe that this has been due in part to the lack of experimentation with layered networks having interior elements capable of active search and reinforcement learning.

8. Discussion

Although the problem and network presented in this article are relatively simple, we believe the approach they illustrate can be applied to more complex nonlinear control problems. Many control problems, both engineering and biological, have the property that the control surface cannot be specified in a simple manner for the entire control space. In these cases it is useful to divide the control space into regions, or control situations, and specify local control surfaces for each region (e.g., Mendel and McLaren, 1970). When possessing a means for accessing the appropriate local control surface for each control decision that must be made, these types of systems can solve complex control problems. This situational control approach is useful when the dynamical characteristics of the controlled system or the objectives of the controller can change drastically, such as, for example, when one changes from driving a car forward to driving it in reverse (turning the wheel clockwise then turns the car left instead of right).

The utility of this approach is made especially apparent by certain theories in artificial intelligence and psychology such as Minsky's theory of "frames" (1975) or Arbib's theory of "schemas" (1978, 1981). These theories suggest that when one encounters a new situation or shifts one's viewpoint, a structure is selected that contains general information about how to act and what to expect in that situation. This structure is parameterized by the situation's particular details. We cannot claim to have implemented a system of frames or schemas, but we do believe that in the example we have presented one can begin to see how such an organization might be learned by a system of neuron-like components using the principles of adaptive representation development, enforcement of variety via lateral inhibition, active search, and reinforcement learning.

4 Some success has been achieved with layered networks capable of "unsupervised" clustering of their inputs into groups of similar patterns (e.g., Fukushima's "neocognitron", 1980). However, this process should not be confused with reinforcement learning. The correctness of a particular clustering is determined solely by the initial representation and does not rely on any form of environmental feedback. There is no functional verification

Appendix A

Nonlinearity of the Landmark Learning Task

We shall prove that the landmark learning task in the environment shown in Fig. 2A is nonlinear in terms of the seven input variables representing proximity to the four landmarks, presence in region A, presence in region B, and a constant input. Even though it is not provided to our network, we include a constant input in our analysis to show that the task would still be nonlinear if this input were available. In Minsky and Papert's (1969) terms, we shall prove that this is not an "order 1" problem. Although their "group invariance theorem" applies to this problem, we prove it in a more direct fashion.

It is clear that the variables corresponding to the circular landmarks are not involved in the problem's nonlinearity since these landmarks are in the same relative positions in the two regions. We are therefore able to omit these variables from our analysis without loss of generality.

Consider any four points P_1, P_2, P_3, and P_4 in the environment shown in Fig. 2A such that P_1 and P_2 in region A are located along the horizontal line connecting the boxes, with P_1 the same distance from the shaded box as P_2 is from the hollow box; and P_3 and P_4 are in the same positions but in region B. The vectors of sensory input at these points are:

$$P_1 = (x, y, 1, 0, 1)$$

$$P_2 = (y, x, 1, 0, 1)$$

$$P_3 = (y, x, 0, 1, 1)$$

$$P_4 = (x, y, 0, 1, 1),$$

where the first component is the input due to the shaded box; the second is due to the hollow box; a 1 in the third position indicates region A; a 1 in the fourth position indicates region B; and where the constant input in the fifth position is set, without loss of generality, to 1. Let the input pattern at point P_i be denoted by the vector $(x_1^i, \ldots, x_5^i)$. If the problem is to be solved, then the action "move east" must be associated with points P_1 and P_3, and the action "move west" must be associated with points P_2 and P_4. Again we can ignore north/south actions and consider actions that are ordered pairs of real numbers. We assume that action (E, W) means "move east" if $E > W$ and "move west" if $W > E$. Then, if the problem can be solved linearly, there exist constant vectors $A = (a_1, \ldots, a_5)$ and $B = (b_1, \ldots, b_5)$ such that

$$\sum_{j=1}^{5} a_j x_j^i > \sum_{j=1}^{5} b_j x_j^i$$

for $i=1$, 3 since the correct movement is east from points P_1 and P_3; and

$$\sum_{j=1}^{5} a_j x_j^i < \sum_{j=1}^{5} b_j x_j^i$$

for $i=2$, 4 since the correct movement is west from points P_2 and P_4.

Writing these inequalities explicitly for points P_1 and P_4, we require

$$(a_1x+a_2y+a_3+a_5)>(b_1x+b_2y+b_3+b_5), \quad \text{(A1)}$$

$$(a_1x+a_2y+a_4+a_5)<(b_1x+b_2y+b_4+b_5). \quad \text{(A2)}$$

Adding a_4 to both sides of (A1) and a_3 to both sides of (A2) yields

$$(b_1x+b_2y+b_3+b_5+a_4)<(a_1x+a_2y+a_3+a_4+a_5)$$
$$<(b_1x+b_2y+b_4+b_5+a_3).$$

Hence,

$$b_3+a_4<b_4+a_3. \quad \text{(A3)}$$

Similarly, the inequalities for points P_2 and P_3 are

$$(a_1y+a_2x+a_3+a_5)<(b_1y+b_2x+b_3+b_5), \quad \text{(A4)}$$

$$(a_1y+a_2x+a_4+a_5)>(b_1y+b_2x+b_4+b_5). \quad \text{(A5)}$$

Adding a_4 to both sides of (A4) and a_3 to both sides of (A5) yields

$$(b_1y+b_2x+b_4+b_5+a_3)>(a_1y+a_2x+a_3+a_4+a_5)$$
$$<(b_1y+b_2x+b_3+b_5+a_4).$$

Hence,

$$b_4+a_3<b_3+a_4$$

which contradicts (A3). Therefore, the problem cannot be solved linearly in terms of the representations we have assumed.

Appendix B

Details of the Simulation Experiment

Inputs. Landmark input values range from 0.0 to 1.0. Region A and region B inputs are either 0.0 or 0.5. Attractant signal values range from 0.0 to 1.0.

Layer 1. The fixed weights have values 1.0. The random variables NOISE_i^1 in Eq. (1) have mean zero normal distributions with standard deviations of 0.1. Layer 1 learning rate constant $c_1=4.0$.

Layer 2. The random variables NOISE_i^2 in Eq. (2) have mean zero normal distributions with standard deviations of 0.1. Layer 2 learning rate constant $c_2=1.0$. $\varepsilon=0.0001$ in Eq. (3). Each spatial step is either 0 or 5 pixels in each direction.

Training Procedure. 1) 360 time steps in region A (Fig. 3A–C). 2) 340 time steps in region B (Fig. 3D–F). 3) 250 time steps in region A (Fig. 3G–I). 4) Test of learning by 100 time steps in region A and 100 time steps in region B (Fig. 3J).

Note. Learning was prevented from occuring for the first time step after the network was removed form one region and placed in another. This prevented any changes in attractant level resulting from these manipulations (rather than from the network's actions) from influencing the connection weight values.

Acknowledgements. This research was supported by the Air Force Office of Scientific Research and the Avionics Laboratory (Air Force Wright Aeronautical Laboratories) through contract F33615-80-C-1088. The authors wish to thank A. H. Klopf, D. N. Spinelli, and E. Hudlicka for their criticisms and contributions.

References

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

Amari, S., Arbib, M.A.: Competition and cooperation in neural nets. In: Systems neuroscience. Metzler, J. (ed.). New York: Academic Press 1977

Arbib, M.A.: Segmentation, schemas and cooperative computation. In: Studies in mathematical biology. Part I. Cellular behavior and the development of pattern. Levin, S. (ed.). Math. Ass. Am. 1978

Arbib, M.A.: Perceptual structures and distributed motor control. In: Handbook of physiology. Motor control. Vol. III. Brooks, V.B. (ed.). Bethesda, MD: Am. Physiol. Soc. 1981 (to appear)

Barto, A.G., Sutton, R.S.: Landmark learning: an illustration of associative search. Biol. Cybern. **42**, 1–8 (1981a)

Barto, A.G., Sutton, R.S.: Goal-seeking components for adaptive intelligence: an initial assessment. Technical Report AFWAL-TR-81-1070, Avionics Laboratory, Air Force Wright Aeronautical Laboratories, Wright-Patterson Air Force Base, Ohio (1981b)

Barto, A.G., Sutton, R.S., Brouwer, P.: Associative search network: a reinforcement learning associative memory. Biol. Cybern. **40**, 201–211 (1981)

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

Fukushima, K.: A model of associative memory in the brain. Kybernetik **12**, 58–63 (1973)

Fukushima, K.: Neccognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol. Cybern. **36**, 193–202 (1980)

Grossberg, S.: Adaptive pattern classification and universal recoding. I. Parallel development and coding of neural feature detectors. Biol. Cybern. **23**, 121–134 (1976a)

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

Ivakhnenko, A.G.: Polynomial theory of complex systems. IEEE Transactions on Systems, Man, and Cybernetics SMC-**1**, 364–378 (1971)

Kilmer, W.L., McCulloch, W.S., Blum, J.: A model of the vertebrate central command system. Int. J. Man-Machine Stud. **1**, 279–309 (1969)

Klopf, A.H., Gose, E.: An evolutionary pattern recognition network. IEEE Trans. Syst. Sci. Cybern. **5**, 247–250 (1969)

Klopf, A.H.: Brain function and adaptive systems – a heterostatic theory. Air Force Cambridge Research Laboratories Research Report AFCRL-72-0164, Bedford, MA, 1972. (A summary appears in: Proc. Int. Conf. Syst., Man, Cybern., IEEE Syst. Man, Cybern. Soc. Dallas, Texas, 1974)

Klopf, A.H.: Goal-seeking systems from goal-seeking components: implications for AI. Cogn. Brain Theor. Newslett. **3**, 2 (1979)

Klopf A.H.: The hedonistic neuron: a theory of memory, learning, and intelligence. Washington, D.C.: Hemisphere Publishing Corp. 1982 (to be published)

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

Michie, D., Chambers, R.A.: BOXES: an experiment in adaptive control. Machine intelligence 2, pp. 137–152. Dale, E., Michie, D. (eds.). Edinburgh: Oliver and Boyd 1968

Minsky, M.L.: Steps toward artificial intelligence. Proc. IRE **49**, 8–30 (1961)

Minsky, M.L.: A framework for representing knowledge. In: The psychology of computer vision, pp. 211–277. Winston, P.H. (ed.). New York: McGraw-Hill 1975

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

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

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

Raibert, M.H.: A model for sensorimotor control and learning. Biol. Cybern. **29**, 29–36 (1978)

Rosenblatt, F.: Principles of neurodynamics. New York: Spartan Books 1962

Selfridge, O.G.: Pattern recognition and modern computers. In: Proceedings of the 1955 Western Joint Computer Conference, Session on Learning Machines, W.H. Ware Chairman, 91–93 (1955)

Spinelli, D.N.: OCCAM: A computer model for a content addressable memory in the central nervous system. In: The biology of memory. Pribram, K., Broadbent, D. (eds.). New York: Academic Press 1970

Sutton, R.S., Barto, A.G.: Toward a modern theory of adaptive networks: Expectation and prediction. Psychol. Rev. **88**, 135–170 (1981)

von der Malsburg, C.: Self-organization of orientation sensitive cells in the striate cortex. Kybernetik **14**, 85–100 (1973)

Received: September 10, 1981

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