Proc. Natl. Acad. Sci. USA
Vol. 88, pp. 4433–4437, May 1991
Neurobiology

A more biologically plausible learning rule for neural networks

(reinforcement learning/coordinate transformation/posterior parietal cortex/sensorimotor integration/Hebbian synapses)

PIETRO MAZZONI[†‡], RICHARD A. ANDERSEN[†§], AND MICHAEL I. JORDAN[†]

[†]Department of Brain and Cognitive Sciences, and [‡]Harvard–MIT Division of Health Sciences and Technology, Massachusetts Institute of Technology, Cambridge, MA 02139

Communicated by Francis Crick, February 21, 1991 (received for review October 19, 1990)

ABSTRACT **Many recent studies have used artificial neural network algorithms to model how the brain might process information. However, back-propagation learning, the method that is generally used to train these networks, is distinctly "unbiological." We describe here a more biologically plausible learning rule, using reinforcement learning, which we have applied to the problem of how area 7a in the posterior parietal cortex of monkeys might represent visual space in head-centered coordinates. The network behaves similarly to networks trained by using back-propagation and to neurons recorded in area 7a. These results show that a neural network does not require back propagation to acquire biologically interesting properties.**

Recently neural network models have been used to model and predict certain aspects of brain function. A criticism of such models, however, has been their reliance on back propagation, a learning algorithm that has been considered "unbiological" because it requires passage of information backward through synapses and along axons and because it uses error signals that must be precise and different for each neuron in the network. Attempts to implement more biologically realistic forms of back-propagation still require unrealistic conditions, such as symmetrical feedback pathways that are identical in every way, including strength of the individual synaptic connections (1, 2). Crick has suggested that "what is really required is a brain-like algorithm which produces results of the same general character as back-propagation" (3).

In our laboratory, we have been refining our neural network models to bring them more in line with what we know of nervous system function. This paper describes the application of a variant of the associative reward-penalty (A_{R-P}) learning rule of Barto and colleagues (4–6) to the training of a multilayer neural network in a biologically relevant supervised learning task. We used this network to model the process of coordinate transformation, believed to be computed by the posterior parietal cortex (for review, see ref. 7) and found that units in the middle layer of the network develop response properties similar to those of area 7a neurons. These properties are also similar to those obtained with a previous model due to Zipser and Andersen (8), which relied on back-propagation learning. The A_{R-P} rule has the advantage of possessing several more physiological correlates, such as a feedback system that transmits performance information along explicit and plausible pathways, Hebb-like synapses that correlate pre- and postsynaptic activity, and a single scalar performance evaluation that is computed from the overall output of the network and is sent to all connections in the network in the form of a reinforcement signal.

MODEL

Neurons in area 7a appear to compute head-centered locations of visual stimuli by combining retinal and eye-position information (7, 9). A feature of the responses of these neurons that may be crucial for this computation is an approximately planar modulation by eye position of the response to a visual stimulus (10). In other words, if one records from an area 7a neuron in an awake monkey while a spot of light is presented at a fixed location on its retina, then as the animal looks in various directions, the neuronal firing rate varies approximately linearly with changes in the horizontal and/or vertical angle of gaze. A plot of this modulation of visual response by eye position is termed the "spatial gain field." Andersen and colleagues (7–10) hypothesized that an ensemble of neurons with this response property, each with its own slope, direction, and range of planar eye position sensitivity, could encode a distributed representation of craniotopic locations. Zipser and Andersen (8) set up a three-layer network to transform the coordinates from a retinotopic frame to a craniotopic one, using retinal-stimulus location and eye position as input signals and the resulting head-centered location as the training signal. After training this network by back-propagation, the units in the middle layer (so-called "hidden" units) displayed planar gain fields remarkably similar to those of area 7a neurons. This result suggested that some fundamental computational feature embodied by the network may be shared by area 7a neurons in their representation of head-centered space.

As properties of the hidden units in the Zipser and Andersen model suggested a possible connection between that model and area 7a, it was natural to ask how crucial back-propagation is for the development of these properties. We addressed this question by training a neural network with an architecture similar to the Zipser and Andersen model but using the more biologically plausible A_{R-P} learning algorithm. Our present network has a three-layer, fully connected, feed-forward architecture (Fig. 1*a*). The input layer consists of a visual and an eye position group of units, which were modeled according to characteristics of area 7a neurons established in previous studies (Fig. 1 *b–c*; ref. 8). The hidden and output layers consist of binary stochastic elements (Fig. 1*d*), which produce an output of 1 with a probability given by the logistic function of the summed weighted inputs and an output of 0 otherwise. The output layer encodes the craniotopic location that is the vector sum of the retinal and eye position inputs and is composed of one of two alternative formats (Fig. 1 *e–f*), one analogous to the monotonic eye position representation and the other to the retinal gaussian format.

We modified the supervised learning procedure for A_{R-P} networks, introduced by Barto and Jordan (6), to train our network. Every unit in the network receives a scalar reinforcement signal r (Fig. 1*a*), the value of which depends on how close the current network output is to the desired output. [Specifically, r assumes a value between 0 and 1, with 0 indicating maximum error in the output (i.e., every unit that should be firing is not and vice versa) and 1 corresponding to

The publication costs of this article were defrayed in part by page charge payment. This article must therefore be hereby marked "*advertisement*" in accordance with 18 U.S.C. §1734 solely to indicate this fact.

Abbreviation: A_{R-P}, associative reward-penalty.

[§]To whom correspondence should be addressed.

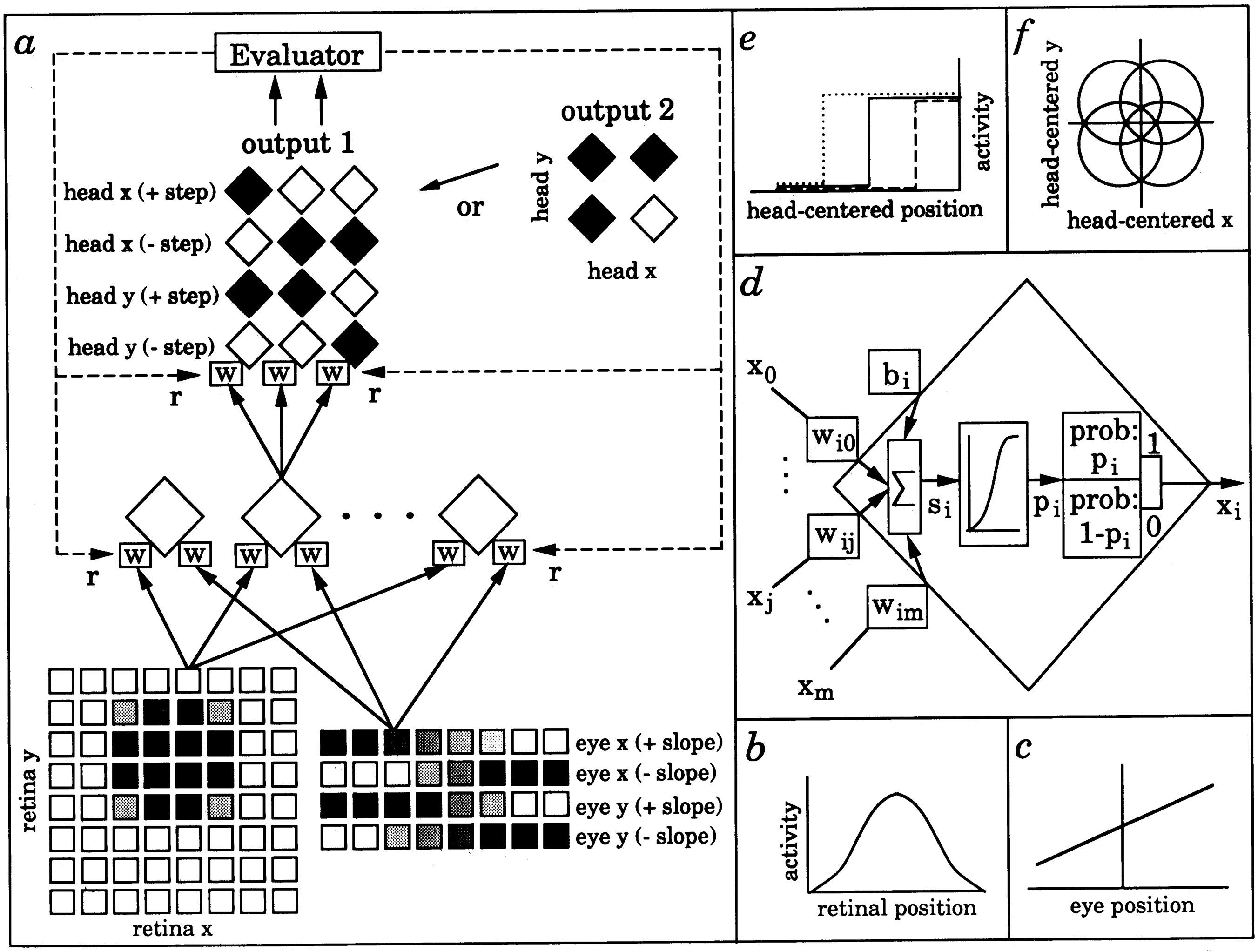

FIG. 1. (*a*) Network structure. Retinal input is encoded by 64 units with gaussian receptive fields (*b*), while eye position is represented by 32 units with linear activation functions (*c*). In the retinal input, each unit has an output between 0 and 1, a 1/*e* width of 15° and a receptive field peak 10° apart from that of its horizontal and vertical neighbors. In the eye-position input, the output of each unit, between 0 and 1, is a linear function of horizontal or vertical orbital angle, with random slope and intercept. These input formats reproduce properties of certain area 7a neurons that respond only to visual stimuli or to changes in eye position. The shading of each unit is proportional to its activity, with black representing maximum activity. The hidden and output layers are composed of binary stochastic elements (*d*), which produce an output of 1 with probability (prob) p equal to the logistic function of the sum of the weighted inputs ($s_i = \Sigma_{j=0}^{m} w_{ij}x_j$), and zero with probability $1 - p$. The *jth* unit in the network provides input x_j to the *ith* unit via the connection w_{ij}; m is the number of inputs to the units, and b is a bias. The network used from two to eight hidden units. The output units encode head-centered locations according to one of two output formats. In the "binary-monotonic" format (*e*), each unit produces an output of 1 or 0, depending on whether the encoded location is to the right or to the left (or, for some units, above or below) a certain reference point. For example, a typical output layer consisted of four sets of three units, giving an output of 1 when the x (or y) craniotopic coordinate is > (or <) −40, 0, or +40 degrees. This format is analogous to the eye-position input format, in that four groups of units encode an increase in horizontal or vertical position angle by increasing or decreasing their activation monotonically. Another format we used is the "binary-gaussian" one (*f*), in which four units give an output of 1 when the spatial position is within 100° of their receptive field centers, which are located at (±60, ±60)°. This format is analogous to that of the retinal input, in that a position angle is encoded topographically by units with overlapping receptive fields.

optimal performance (no error in the computed head-centered position).] The weights of all the connections are then adjusted, after each pattern presentation, in such a way as to maximize the value of this reinforcement. If we let x_i denote the output of the *i*th unit in the network, p_i denote its probability of firing, and w_{ij} denote the connection weight for its input from the *j*th unit (Fig. 1*d*), the equation for updating the weights is

$$\Delta w_{ij} = \rho r(x_i - p_i)x_j + \lambda\rho(1 - r)(1 - x_i - p_i)x_j, \qquad [1]$$

where ρ and λ are constants. The first term in this sum computes the reward portion of the learning rule, whereas the second term is the penalty portion. Ignoring for the moment the constant terms and the stochastic component, this equation changes the synaptic weights by correlating the reinforcement signal, presynaptic activity, and postsynaptic activity. Thus, for instance, a correct response (large r) will strengthen connections that were active during the response, and an incorrect response (small r) will weaken active synapses. The value of r is a function of the average output error and is computed as $r = 1 - \varepsilon$, with

$$\varepsilon = \left\{ \frac{1}{K} \sum_{k=1}^{K} |x_k^* - x_k| \right\}^{1/n}, \qquad [2]$$

where k indexes the K output units in the network, x_k^* is the desired output of the kth unit in the output layer, x_k is its actual output, and n is a constant¶.

¶A bias b_i on each unit is also adjusted (as described in ref. 11) by the

RESULTS

The A_{R-P} network learned to perform the coordinate transformation task to any desired accuracy. Fig. 2*b* shows the general behavior of the A_{R-P} network, as it learns to transform 12 pairs of retinal and eye positions into craniotopic coordinates. The learning curve of a corresponding back-propagation network, using the same training set, is shown in Fig. 2*a*. The learning curve of the A_{R-P} network is much noisier than that of back-propagation due to the stochastic nature of its hidden units and to the type of error signal used in A_{R-P} training. The two curves, however, have similar envelopes and the times required for convergence are comparable.[||]

One interesting feature of artificial neural networks is their ability to discover general solutions; once they have learned from a particular set of examples, they can also produce reasonably correct outputs for inputs that the network has never experienced. We tested our network for two types of generalization abilities. In one task we presented it with a set of new, random input pairs of retinal location and eye position that coded for the same output locations as the original training set. As shown in Fig. 3 (*i*), both back-propagation and A_{R-P}-trained networks performed this task extremely well. The other generalization task required the trained networks to give the correct output for input patterns coding for new output locations, which is a more difficult task. Although both networks produced some error (Fig. 3 *ii*), it was still considerably less than for the untrained ones, indicating that both networks generalized to a reasonable extent.

Using a similar approach to the one we used in neurophysiological experiments, we examined the dependence of the activity of the hidden units on two parameters, eye position and retinal stimulus location. Spatial gain fields were obtained by holding retinal position constant and varying eye position, whereas visual receptive fields were obtained by holding eye position constant and varying retinal position (Fig. 4). In both cases we did not measure the instantaneous output of the unit itself (which is binary) but its probability of firing (a continuous variable). As Fig. 4 shows, both the gain fields and the receptive fields of the various hidden units of the network bear a qualitative similarity to those of area 7a neurons. The degree of similarity is approximately equivalent to that produced by Zipser and Andersen's back-propagation-trained network (8). In particular, the gain fields of the hidden units are largely planar in their overall probability of firing (Fig. 4*b*, outside circles), whereas the visually evoked component (dark circles) displays a more variable dependence on eye position. This result was also produced by back-propagation training and found in 78% of spatially tuned area 7a neurons (Fig. 4*a*; ref. 12). These neurons also have unusual receptive fields (Fig. 4*c*; ref. 12), which set them apart from those of many other visual areas. These fields are very large, with diameters extending to 80°, and have complex surfaces, characterized by one or more smooth peaks at various eccentricities. These features were both reproduced by the hidden units of the A_{R-P} network (Fig. 4*d*).

The solutions computed by A_{R-P} training and by back propagation are not just similar in the qualitative sense of the response properties they confer to the hidden units. In fact, we found that for a given training pattern, the set of weights trained by the A_{R-P} algorithm may be transferred to a back-propagation network (with continuous output hidden units but trained with target locations in the binary output format of the A_{R-P} network) without any appreciable reduction in the accuracy of the network response to that training pattern and vice versa. The individual values of the weights of the connections are *not* the same after A_{R-P} and back-propagation training, but the overall structure of these weights is such that the solutions of the two algorithms for the coordinate-transformation problem are functionally equivalent.

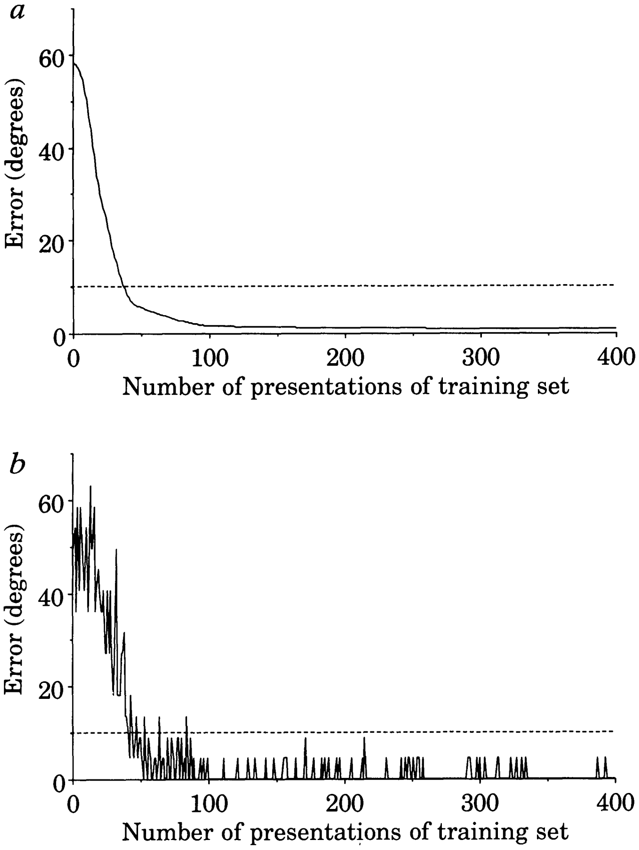

FIG. 2. (*a*) Learning curve for a back propagation network with three hidden units, trained on 12 pairs of retinal and eye positions, each encoding one of four craniotopic locations. Two output units were used, each encoding horizontal or vertical craniotopic location as a linear function of activity, which allowed us to compute the location encoded by the network during training and, thus, record the output error in degrees of visual angle. Graph shows the average, over various training inputs, of the absolute values of horizontal and vertical differences between desired and actual craniotopic coordinates, plotted against number of presentations of the input training set. The dotted line indicates spacing between the receptive field peaks of the retinal input units. Parameter values were 0.1 for learning rate and 0.9 for momentum term (see ref. 11). (*b*) Learning curve for an A_{R-P} network with the same architecture. Values of parameters were $\rho = 0.5$, $\lambda = 0.01$, and $n = 6$. The two output units encoded whether the horizontal or vertical craniotopic location was $>$ or $<$ 0. Average output error was converted into degrees by using the same factor as for the back-propagation network, so that the two curves could be compared.

DISCUSSION

The A_{R-P} rule, like back propagation, trains networks of adaptive elements by adjusting the connection strengths along the direction of the gradient of a predefined performance measure. It does so, however, by computing only an

rule in Eq. **1**. Typical values for the parameters in equations **1** and **2** were $\rho = 0.5$, $\lambda = 0.01$, and $n = 3$.

[||] As the number of epochs becomes large (>1000) the output error of both networks approaches 0. For the back-propagation network, which has a continuous output, the error decreases asymptotically, whereas for the A_{R-P} network, which has a binary output, the error spends increasingly more time at the value 0, flickering occasionally to the value of the smallest resolvable angle of the output. Neither algorithm had serious problems with local minima (frequency of local minima was ≈5% for back propagation and <1% for the A_{R-P} algorithm in ≈200 simulations).

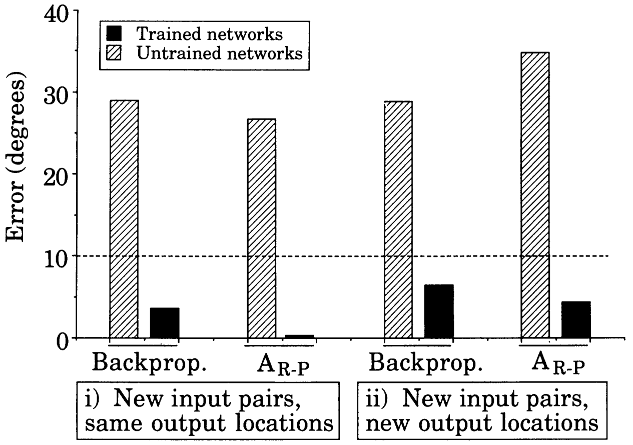

FIG. 3. Output error produced by back-propagation and A_{R-P} networks described in Fig. 2 before and after training with 40 input pairs, when presented with (*i*) 40 new, random inputs coding for the same output locations as in the training set, and (*ii*) 40 random inputs coding for 40 new, random output locations. Error was computed for each network as described in Fig. 2.

estimate of this gradient (6, 13). Units trained by the A_{R-P} rule do not have the detailed information about the error vector and the state of other units that is necessary to compute the exact gradient and which back-propagation units obtain through nonbiological pathways. Due to the random noise in their output, however, A_{R-P} units can "jitter" their activity during learning so as to get an estimate of how variations in activity affect the reinforcement they receive, which, in turn, allows them to estimate the direction in weight space along which to change their weights to increase reinforcement. Although this method allows A_{R-P}-trained units to properly adjust their weights using only locally available information, it is more random in its search for a solution than back-propagation, as reflected in the fluctuations in the learning curve in Fig. 2*b*. The precise computation through back-propagation of the performance gradient tells the algorithm the exact manner in which to change the weights so that the error is monotonically decreased, resulting in the smooth curve of Fig. 2*a*.

An important element of the A_{R-P} model, which aligns it with many neurobiological models of learning, is the reinforcement signal. As in any supervised learning scheme, this signal is computed by comparing the activities of output units to desired activities. After these errors are averaged, however, the feedback system transmits only a single value to all the network connections and is not assumed to provide these connections with separate information about the activities of individual output units. The fact that in A_{R-P} training a single value is valid for all the connection weights implies that only one projection is necessary from the reinforcement computing region to area 7a. The existence of signals originating from a small cluster of neurons distributed to entire cortical areas and that possibly carry information about reward has been suggested by anatomical as well as experimental studies (e.g., see ref. 14). In contrast, back propagation requires as feedback an error vector the components of which must course to the appropriate output units and from there to individual hidden units along specified pathways, either retrogradely along axons or through complicated feedback loops with completely symmetrical connection strengths (1, 2).

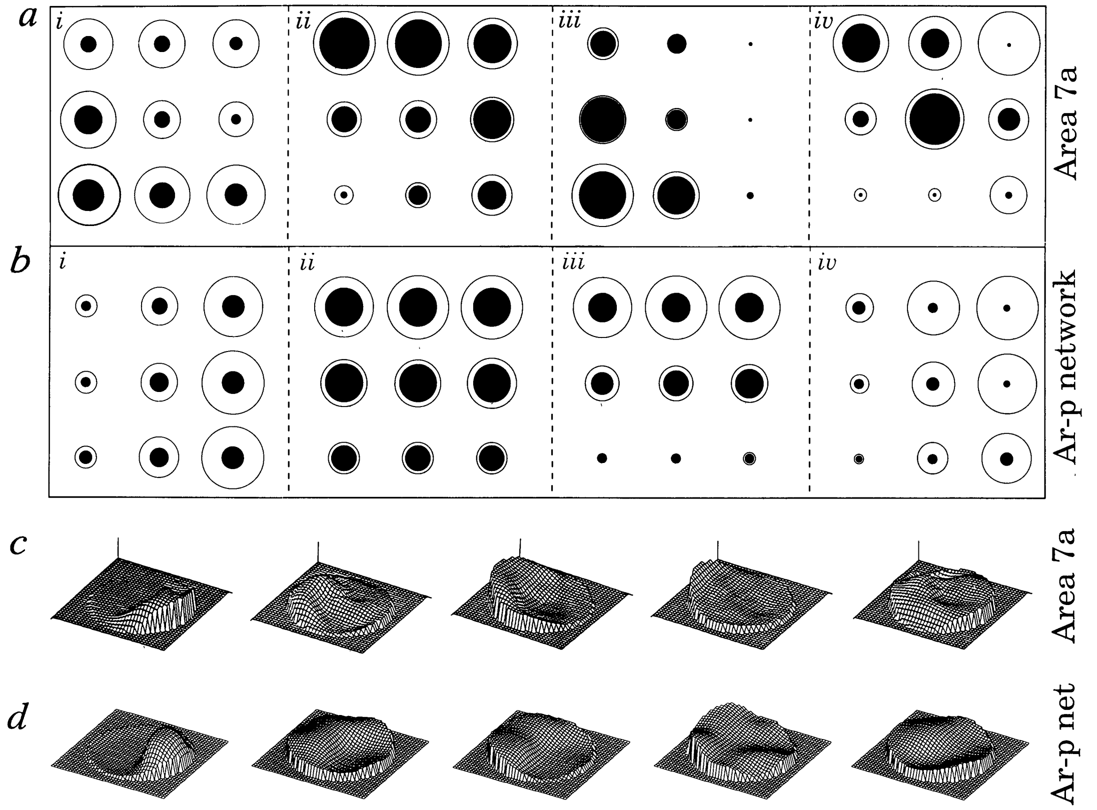

FIG. 4. (*a*) Spatial gain fields recorded from four area 7a neurons (*i–iv*) (8). Outside thin circles represent overall activity elicited by a visual stimulus. Each circle corresponds to one of nine eye positions, spaced 20° apart. The dark circles represent the visual contribution to the response, while the annulus is the eye-position contribution. (*b*) Gain fields of four hidden units in A_{R-P} networks trained on various sets of four pairs of retinal and eye positions. (*c*) Receptive fields of four area 7a neurons (8). The response to a visual stimulus is plotted against the retinal location of the stimulus. (*d*) Receptive fields of four hidden units in A_{R-P} networks trained on various sets of four pairs of retinal and eye positions. Radius of sampling circle is 40° for the plot in *c* and *d*.

Another "biological" feature of learning by A_{R-P} units is the use of information locally available to the synapse the strength of which is being adjusted. The A_{R-P} learning rule (Eq. **1**) is a sum of two terms, each containing the following: (*i*) the reinforcement signal r (and the corresponding penalty value, $1 - r$); (*ii*) information regarding the current state of the unit ($x_i - p_i$); and, (*iii*) the input (x_j) from each unit connecting to this unit. We have already discussed r. The variable x_i is the output of the unit (0 or 1), and p_i is the probability that this output will be 1 given the current net input, which depends on the weights of the unit. The quantity p_i could be interpreted as the rate at which the unit will fire, given the present input. These two values, as well as x_j, are directly available at the connection between the input, or "presynaptic," unit and the given ("postsynaptic") unit. With back-propagation, on the other hand, changes in strength at one connection require information about the activities and error signals for all units in the following layers. The A_{R-P} rule, therefore, embodies a fundamental feature of Hebbian learning—that is, the proportionality of a change in synaptic strength to both presynaptic and postsynaptic signals (ref. 15; for reviews, see refs. 16 and 17). Indeed, the connections in an A_{R-P} network fit a modern definition of Hebbian synapses introduced by Brown and coworkers, in that they embody "a time-dependent, highly local and strongly interactive mechanism to increase synaptic efficacy as a function of the conjunction or correlation between pre- and postsynaptic activity" (16). Hebbian learning remains one of the more plausible mechanisms for synaptic strength modification, both on theoretical (18) and experimental grounds (e.g., refs. 15, 16, and 19). The reinforcement signal in the A_{R-P} rule does not alter the Hebbian character of the algorithm; indeed, it has been suggested that such "global control signals . . . may enable the induction or consolidation of changes at synapses that have met the . . . criteria for a Hebbian modification . . . and thus control Hebbian plasticity in a large population of activated synapses" (16).

The last feature that adds some biological flavor to the A_{R-P} unit is the probabilistic nature of its output. The unpredictability of the exact firing rate produced by a neuron for any given presentation of a certain input has long been recognized as a feature of nerve cells. In fact, this stochastic aspect of activity is one of the reasons neurophysiologists usually present data as summed histograms of several trials (20). This is a feature not included in the deterministic units of back-propagation networks. In the A_{R-P} network, moreover, the noise of the units is an essential component of the learning process, as it produces the variability in the output necessary to direct the search for a solution in an environment that provides limited feedback information. Similarly, the noise in neuronal activity may play an important role in biological learning.

We have not examined in our study the issue of how the A_{R-P} algorithm behaves for networks with considerably larger numbers of hidden units and training locations. We expect learning to be significantly slower for such networks (6). It is possible, however, that the algorithm could be modified to address the scaling issue, for example, by embodying more specificity—perhaps of a topographic nature—in the reinforcement signal. In this sense one could view our use of a single scalar feedback signal as a worst-case scenario that does not exclude more specialized signals that may be used by biological systems.

Overall, we have shown that a number of features of the A_{R-P} algorithm bring it closer than back-propagation to what is known about biological learning. The fact that a more biologically plausible algorithm produces hidden unit response properties like those of area 7a neurons supports the validity of neural-network models as tools for studying the computations by populations of neurons in cortical areas. We must emphasize, however, that the focus of our interest at this point is not in how literally A_{R-P} networks reproduce individual neurophysiological processes; it is rather the fact that the A_{R-P} rule, back-propagation, and perhaps other supervised learning algorithms may form a family of training procedures that yield similar functional representations when applied to parallel networks and that some of these algorithms can do so by using mechanisms not excluded and, perhaps, suggested by neurophysiological evidence.

We thank Sabrina J. Goodman for helpful discussion and for making available several computer programs. This work was supported by grants from the Office of Naval Research (N00014-89-J-1236) and the National Institutes of Health (EY05522) to R.A.A., by a Medical Scientist Training Program Grant (National Institutes of Health 5T32GM07753-10) to P.M., and by a grant from the Siemens Corporation to M.I.J.

1. Zipser, D. & Rumelhart, D. E. (1990) in *Computational Neuroscience*, ed. Schwartz, E. L. (MIT Press, Cambridge, MA), pp. 192–200.
2. Parker, D. B. (1985) Technical Report TR-47 (Center for Computational Research in Economics and Management Science, MIT, Cambridge, MA).
3. Crick, F. H. C. (1989) *Nature (London)* **337**, 129–132.
4. Barto, A. G. (1985) *Hum. Neurobiol.* **4**, 229–256.
5. Barto, A. G. (1989) in *The Computing Neuron*, eds. Durbin, R. M., Miall, R. C. & Mitchison, G. J. (Addison-Wesley, Reading, MA) pp. 73–98.
6. Barto, A. G. & Jordan, M. I. (1987) *Proc. IEEE Int. Conf. on Neural Networks* **2**, 629–636.
7. Andersen, R. A. (1989) *Annu. Rev. Neurosci.* **12**, 377–403.
8. Zipser, D. & Andersen, R. A. (1988) *Nature (London)* **331**, 679–684.
9. Andersen, R. A. (1988) in *Neurobiology of Neocortex*, eds. Rakic, P. & Singer, W. (Wiley, New York), pp. 285–295.
10. Andersen, R. A., Essick, G. K. & Siegel, R. M. (1985) *Science* **230**, 456–458.
11. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1986) in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, eds. Rumelhart, D. E., McClelland, J. L. & PDP Research Group (MIT Press, Cambridge, MA), Vol. 1, pp. 318–362.
12. Andersen, R. A. & Zipser, D. (1988) *Can. J. Physiol. Pharmacol.* **66**, 488–501.
13. Williams, R. J. (1987) *Proc. IEEE Int. Conf. on Neural Networks* **2**, 601–608.
14. Richardson, R. T., Mitchell, S. J., Baker, F. H. & Delong, M. R. (1988) in *Cellular Mechanisms of Conditioning and Behavioral Plasticity*, eds. Woody, C. D., Alkon, D. L. & McGaugh, J. L. (Plenum, New York), pp. 161–173.
15. Kelso, S. R., Ganong, A. H. & Brown, T. H. (1986) *Proc. Natl. Acad. Sci. USA* **83**, 5326–5330.
16. Brown, T. H., Kairiss, E. W. & Keenan, C. L. (1990) *Annu. Rev. Neurosci.* **13**, 475–511.
17. Squire, L. R. (1987) *Memory and Brain* (Oxford Univ. Press, Oxford, U.K.).
18. Linsker, R. (1986) *Proc. Natl. Acad. Sci. USA* **83**, 7508–7512.
19. Sejnowski, T. J., Chattarji, S. & Stanton, P. K. (1989) in *The Computing Neuron*, eds. Durbin, R. M., Miall, R. C. & Mitchison, G. J. (Addison-Wesley, Reading, MA) pp. 105–124.
20. Sejnowski, T. J. (1981) in *Parallel Models of Associative Memory*, eds. Hinton, G. E. & Anderson, J. A. (Erlbaum, Hillsdale, NJ), pp. 189–212.