

# $A_{R-P}$ learning applied to a network model of cortical area 7a

Pietro Mazzoni<sup>1</sup>, Richard A. Andersen<sup>2</sup> and Michael I. Jordan<sup>2</sup>

<sup>1</sup>Harvard-MIT Division of Health Sciences and Technology, and <sup>2</sup>Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA 02139

## Abstract

We describe a neural network that learns to transform retinotopic coordinates of visual stimuli into a head-centered reference frame by combining retinal stimuli with eye position. Area 7a of primate cortex is thought to perform a similar transformation. The neurons involved have unique response properties (planar modulation of visual response by eye position, and large complex receptive fields) and appear to represent head-centered space in a distributed fashion [1]. Our network's architecture is similar to that of a previous backpropagation model of area 7a [2,17], but is trained with a gradient-descent algorithm that is more biologically plausible than backpropagation. This algorithm is a variant of the associative reward-penalty ( $A_{R-P}$ ) learning rule [3,5], and uses a global "reinforcement" signal to adjust the connection strengths. Our network learns to perform the task successfully to any accuracy and generalizes appropriately, and the hidden units develop response properties very similar to those of area 7a neurons. These results show that a learning network does not require backpropagation to acquire biologically interesting properties. These may arise naturally from the network's layered architecture and from the supervised learning paradigm.

## Introduction

One of the most notable features of neural networks is that the manner in which they process information appears in many ways "brain-like." For instance, they can learn a mapping from examples, they can work as content-addressable memories, and their performance degrades gracefully with increasing noise in the input signal. Even their architecture—a large number of highly interconnected, relatively simple elements, collectively capable of complex computations—is reminiscent of popular notions of brain structure. This feature has stimulated many researchers to develop neural network models of biological information processing systems, which in some cases has led to the discovery of biologically interesting features in the networks' style of computation.

An example of this is a neural network model of area 7a of primate parietal cortex developed by Zipser and Andersen [2,17]. Neurons in this region appear to compute head-centered locations of visual stimuli by combining retinal and eye-position information (cf. [1] for a review). A feature of these neurons' responses that may be crucial for this computation is an approximately planar modulation by eye position of the response to a visual stimulus [2,17]. 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 neuron's 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." It was therefore 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 set up a three-layer network to perform the coordinate transformation 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 backpropagation, the 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, such as its layered architecture or its supervised learning paradigm, may be shared by area 7a neurons in their representation of head-centered space.

While the general computational features of neural networks, as well as distinctive ones exhibited by networks such as Zipser and Andersen's, strengthen the possibility of a relationship with biological networks,

the manner in which they learn to process information has generally not been very convincing as a possible model of biological learning. One reason for this is that we know much less about the learning mechanisms employed by biological neural networks than we do about their structure and response properties. Another, perhaps more important reason is that the algorithms used to train artificial neural networks usually involve mechanisms that are far beyond what we consider plausible for a biological system. Implementing the basic version of backpropagation in the nervous system, in particular, would require processes, such as the retrograde propagation of detailed error signals along axons and through synapses, and the adjustment of synaptic strengths by nonlocal computations, that are not accepted as likely candidates for learning mechanisms in the brain. Various approaches to this quandary have been proposed, including modifications of the fundamental architecture of backpropagation networks (e.g., cf. [7]), but a biologically plausible solution remains to be seen. As the properties of the hidden units in the Zipser & Andersen model suggested a possible connection between that model and area 7a, it was natural to ask how crucial is backpropagation for the development of these properties.

We addressed this question by training two neural networks with architecture similar to the Zipser and Andersen model using a supervised learning paradigm that is more plausible from a biological perspective than backpropagation. We describe one of these networks, the “All  $A_{R-P}$ ” network, in this paper (cf. [10] for a more detailed description of both networks). The algorithm we used, which is a variant of the associative reward-penalty ( $A_{R-P}$ ) algorithm for supervised learning introduced by Barto and Jordan [6], trains a neural network by broadcasting a global “reinforcement,” or payoff, signal to all the network’s connections. Our network learned to perform the coordinate transformation task to any desired accuracy, and generalized appropriately. Moreover, the hidden units acquired response properties very similar to those of area 7a neurons, as they did in the Zipser and Andersen model.

### Network Structure And Training

Our network has a three-layer, fully-connected, feed-forward architecture (fig. 1a). The input layer consists of a visual (sensory) and an eye position (proprioceptive) group of units, which were modelled according to characteristics of area 7a neurons established in previous studies (fig. 1b-c) [17]. The hidden and output layers consist of binary stochastic elements (fig. 1d), which produce an output of one with probability given by the logistic function of the summed weighted inputs, and an output of zero 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, one analogous to the monotonic eye position representation, and the other to the retinal gaussian format. Because all the connection strengths in our network are adjusted by the  $A_{R-P}$  learning rule (see below), we refer to it as the *All  $A_{R-P}$  network*.

We modified the supervised learning procedure for  $A_{R-P}$  networks, introduced by Barto and Jordan [6], to train our network. The input pattern is a signal for the retinal location of a visual stimulus paired with one for the current eye position. The desired output pattern is one that codes for a head-centered location that is the vector sum of the retinal and eye positions. The essence of the algorithm is the  $A_{R-P}$  learning rule. Every binary stochastic element in a given network receives a scalar payoff (or reinforcement) signal  $r$  (fig. 1a), whose value, in the supervised learning paradigm, depends on how close the current output is to the desired output. Specifically, assumes a value between 0 and 1, with 0 indicating maximum error in the output angle (in our network, an output angle more than 80 degrees off target), and 1 corresponding to optimal performance (no error in the computed head-centered position). The weights of the input connections on each binary stochastic element are then adjusted, after each pattern presentation, in such a way as to maximize the value of this payoff. If we let  $x_i$  represent the output of the  $i^{th}$  unit in the network,  $p_i$  its probability of firing, and  $w_{ij}$  the connection weight for its input from the  $j^{th}$  unit (fig. 1d), the equation for updating the weights is

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

where  $\rho$  and  $\lambda$  are constants. The bias  $b_i$  on each unit is also adjusted by the rule in eq.1. Typical values for the parameters in this equation were 0.3 for  $\rho$  and 0.01 for  $\lambda$ . We will comment on this equation in more detail in the Discussion. The value of  $r$  is computed as:  $r = 1 - \epsilon$ , and

$$\epsilon = \left\{ \frac{1}{K} \sum_{k=1}^K |y_k^* - y_k| \right\}^{\frac{1}{n}} \quad (2)$$

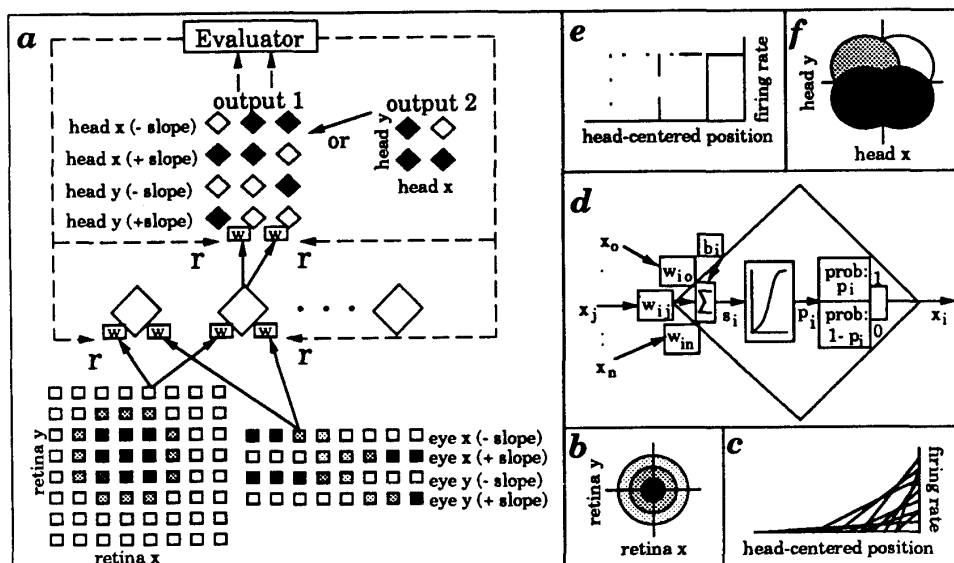


Figure 1: (a) Network structure. Retinal input is encoded by 64 units with gaussian receptive fields (b), while eye position is represented by 32 units with linear activation functions (c). (d) Binary stochastic element. The output units encode head-centered locations according to either a “binary-monotonic” format (e), in which 4 triplets of units give an output of 1 if the  $x$  (or  $y$ ) coordinate is greater (or less) than  $-40$ ,  $0$  or  $+40$  degrees, or according to a “binary-gaussian” format (f), which consists of 4 units giving an output of 1 when the spatial position is within 100 degrees of their receptive field centers. See [10] for details.

where  $k$  indexes the  $K$  output units in the network,  $y_k^*$  is the desired output of the  $k^{th}$  unit in the output layer, and  $y_k$  is its actual output. Values for  $n$  ranged from 2 to 6. This expression for  $\epsilon$  is slightly different from the one used by Barto and Jordan [6], who computed  $\epsilon$  as the sum of the squares of the output units' errors. Both expressions give a quantity nonlinearly related to the error's absolute value, but ours is more sensitive to small errors (as the absolute error for a given unit is always less than or equal to one). Following Barto and Jordan, we refer to this learning rule as the “S-model  $A_{R-P}$  rule.”

## Results

The  $A_{R-P}$  network learned to perform the coordinate transformation task to any desired accuracy. Fig. 2a shows the network's general behavior during training and compares it to that of a backpropagation network learning from the same training set (fig. 2b). The  $A_{R-P}$  network's learning curve is much noisier than backpropagation's, due to the stochastic nature of their hidden units and to the type of error signal used in  $A_{R-P}$  training (see Discussion). The two curves, however, have similar envelopes, and the times required for convergence are comparable. As the number of epochs becomes large ( $> 1000$ ) the output error of both networks approaches 0. For the backpropagation network, which has a continuous output, the error decreases asymptotically, while for the  $A_{R-P}$  network, which has a binary output, the error spends increasingly more time at the value 0, flickering ever so seldom to the value of the output's smallest resolvable angle. Neither algorithm had serious problems with local minima (the frequency of local minima was around 5 percent for backpropagation, and less than 1 percent for the  $A_{R-P}$  algorithm, in approximately 200 simulations).

We tested our network for two types of generalization abilities. One is the ability to perform the correct vector addition of random new input patterns that code for one of the fixed spatial locations in the training set. As shown in fig. 2c, both backpropagation- 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

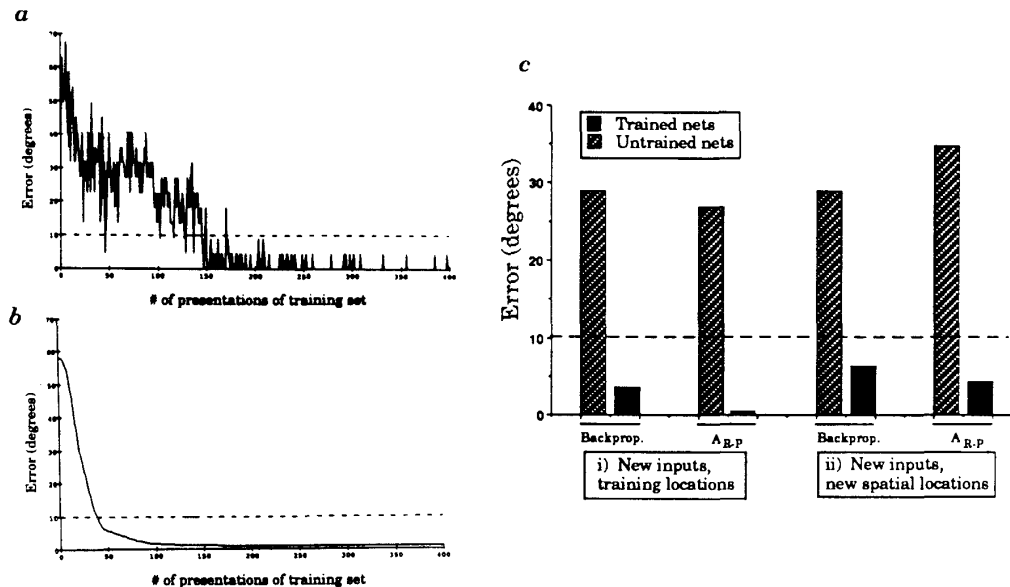


Figure 2: (a) Learning curve for the  $A_{R-P}$  network. The difference between the angular spatial position coded by the output layer and the desired one is plotted against the number of presentations of the training input set. The dotted line indicates the retinal input's resolution. (b) Learning curve for a backpropagation network of similar structure. (c) Output error produced by backpropagation and  $A_{R-P}$  networks when presented with (i) 100 random inputs coding for the same spatial locations as in the training set, and with (ii) 100 random inputs coding for 100 random spatial locations.

input patterns coding for new spatial locations, which is a more difficult task. Although both networks produced some error, it was still considerably less than for the untrained nets, indicating that the networks generalized to a reasonable extent.

Following the experimental approach [2,17], we examined the dependence of the hidden units' activity on two parameters, eye position and retinal stimulus location, obtaining spatial gain fields and visual receptive fields, respectively (fig. 3). In both cases we did not measure the unit's instantaneous output itself (which is binary), but its probability of firing (a continuous variable). As fig. 3 shows, both the gain fields and the receptive fields of the network's various hidden units bear a qualitative similarity to those of area 7a neurons. The degree of similarity is approximately equivalent to that produced by Zipser & Andersen's backpropagation-trained network. In particular, the hidden units' gain fields are largely planar in their overall probability of firing (fig. 3b, outside circles), while the visually evoked component (dark circles) displays a more variable dependence on eye position. This result was also produced by backpropagation training [17], and found in 78 percent of spatially tuned area 7a neurons (fig. 3a; [2]). These neurons also have unusual receptive fields (fig. 3c; [2,17]), which sets them apart from those of most other visual areas. They are very large, with diameters extending to 80 degrees, and have complex surfaces, characterized by one or more smooth peaks at various eccentricities. These qualitative features were both reproduced by the  $A_{R-P}$  network's hidden units (fig. 3d).

The solutions computed by S-model  $A_{R-P}$  training and by back-propagation are not just similar in the qualitative sense depicted in fig. 3. 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 and same output format) without any appreciable reduction in the accuracy of the network's response to that training pattern, and vice versa (cf. [10] for details). The individual values of the weights are not the same after training by the two procedures, but their distribution over the network is such that their overall solutions to the coordinate transformation problem are quantitatively equivalent.

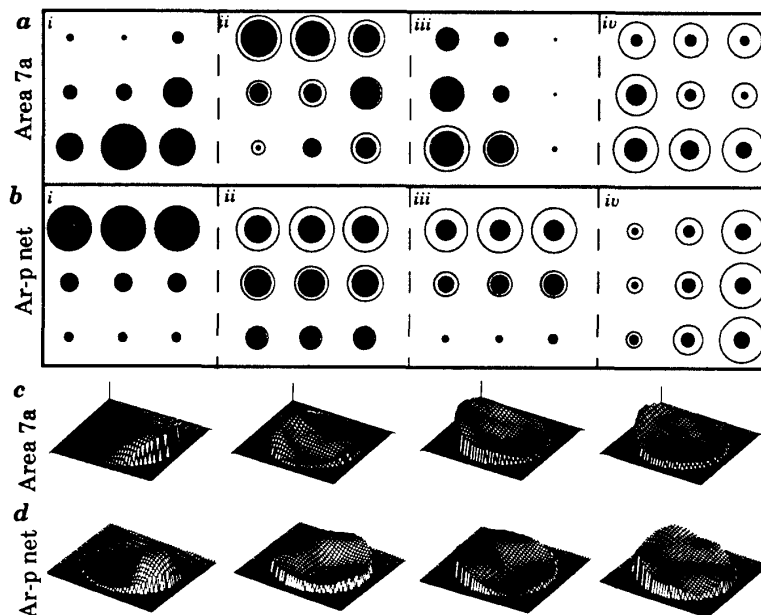


Figure 3: (a) Spatial gain fields recorded from 4 area 7a neurons (i-iv) [17]. The outside thin circles represent the overall activity elicited by a visual stimulus. Each circle corresponds to one of nine eye positions, spaced 20 deg. apart. The dark circles represent the visual contribution to the response, while the annulus is the eye position contribution. (b) Gain fields of 4 hidden units in the  $A_{R-P}$  network. (c) Receptive fields of 4 area 7a neurons [17]. The response to a visual stimulus is plotted against the stimulus' retinal location. (d) Receptive fields of 4  $A_{R-P}$  network hidden units.

### Discussion

The S-model  $A_{R-P}$  algorithm, like backpropagation, trains networks of adaptive elements by adjusting the connection strengths along the direction of the performance measure's gradient. While backpropagation, however, computes the exact value of the error's gradient for a given input pattern, the  $A_{R-P}$  rule computes only an estimate of that gradient [6,16]. Units trained by the  $A_{R-P}$  rule do not have the detailed information about the error vector and the state of other units which is necessary to compute the exact gradient and which backpropagation units obtain through non- biological 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 payoff they receive, which in turn allows them to estimate the direction in weight space along which to change their weights in order to increase reinforcement. While 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 backpropagation, as reflected in fig. 2a. Backpropagation's precise computation 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. 2b.

A crucial requirement for our choice of a learning algorithm was a greater plausibility of biological implementation than backpropagation. We must point out at the outset, however, that  $A_{R-P}$  networks were not designed as literal models of biological neural networks. Because of the poor knowledge we have of the mechanisms of information processing used by the nervous system, the most useful connection between artificial and biological neural networks is presently limited to the description of abstract processes in simplified models and the investigation of the possibility of implementation in the biological hardware. In other words, the  $A_{R-P}$  element was not designed by collecting scattered known facts of neurobiology and molding them into a computationally interesting unit capable of supervised learning, but rather as a sim-

ple, "neurally inspired" element with a few theoretically motivated features that give it interesting learning abilities. We will discuss biological plausibility, therefore, in its literal sense of suggesting that the abstract computing processes performed by the  $A_{R-P}$  unit during learning are more in keeping with possible neural mechanisms proposed and partly demonstrated by experimental neuroscientists than the mechanisms used by back-propagation networks (cf. also [3,4]).

The first and most important element of  $A_{R-P}$  models which aligns them with many neurobiological models of learning is the scalar payoff signal. This has the attractive features of being computed from an average of the error of the output units, and of being transmitted as a single value to all the units in the network. In an analogy with how an animal may learn the coordinate transformation task, the input pattern would correspond to a visual stimulus seen with the eyes at a known angle of gaze (sensed by proprioceptive or corollary discharge pathways). The animal may then guess where the stimulus is in head-centered space by foveating it, and any discrepancy could be used to generate an error signal. This error could be detected as a function of the angular difference between an object in space and the end position of a reaching arm movement to that object. After successful training, this difference would be nil and reinforcement would be maximal. The reinforcement signal could thus be computed by a part of the nervous system that monitored the animal's behavior with no information about the activity of area 7a neurons. In contrast, backpropagation requires as feedback an error vector that takes into account the actual and desired activities of each output unit and compares them individually. The fact that in  $A_{R-P}$  training a single value is valid for all the units implies that only one connection is necessary from the reinforcement computing region to area 7a. The existence of signals originating from a small cluster of neurons and distributed to entire cortical areas has been suggested by anatomical as well as experimental studies (e.g., cf. [12]). In the backpropagation algorithm, on the other hand, the individual error signal must course from the output layer to individual units in the hidden layers along specified pathways, either retrogradely along axons or through rather complicated feedback loops [11].

Another "biological" feature of learning by  $A_{R-P}$  units is the use of information that is locally available to the synapse whose strength is being adjusted. The  $A_{R-P}$  learning rule (eq. 1), is composed of two main terms (in square brackets), one assigning the "reward" part and the other the "penalty" part of the associative reward-penalty concept. Ignoring the constants, the components of these terms are: i) the payoff 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  $r$ .  $x_i$  is the unit's output (0 or 1), and  $p_i$  is the probability that the unit's output will be one given the current net input, which depends on the unit's weights.  $p_i$  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. The  $A_{R-P}$  rule therefore embodies one of the most important elements of Hebbian learning, that is, the proportionality of a change in synaptic strength to both presynaptic and postsynaptic information. Hebbian learning remains one of the more plausible mechanisms for synaptic strength modification, both on theoretical [9] and experimental grounds (e.g., [15]). With backpropagation, on the other hand, changes in strength at one connection require information about the activities and error signals for all the units in the following layers.

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 [14]. This is a feature that is not included in the deterministic units of backpropagation networks.

## Conclusion

Our results represent a step toward establishing the validity, from a physiological perspective, of parallel-architecture learning networks with distributed representations as models of cortical area 7a. We have shown that the  $A_{R-P}$  algorithm can train a network with the Zipser and Andersen model's structure to perform a coordinate transformation task, producing solutions that give hidden unit response properties (planar gain fields and large visual receptive fields) very similar to those of area 7a neurons presumed to code for spatial location. These properties, therefore, are not a specific result of backpropagation, but seem rather to arise from the network's layered architecture and supervised learning paradigm. The fact that backpropagation

and  $A_{R-P}$  compute equivalent solutions for the coordinate transformation task implies that the details of implementation of the learning procedure are not important from a computational point of view. The ability of networks trained with either algorithm to generalize appropriately from a small training set shows that these networks indeed learn to perform coordinate transformations, and do not merely act as content-addressable memories.

We have shown that an  $A_{R-P}$  network can learn to compute coordinate transformations, and that in doing so its hidden units develop gain fields and receptive fields qualitatively similar to those of area 7a neurons. We have also pointed out a number of features of the  $A_{R-P}$  algorithm that bring it closer than backpropagation to what is known about biological learning. We must emphasize again, however, that the focus of our interest at this point is not in how literally  $A_{R-P}$  nets reproduce individual neurophysiological processes. It is rather on the fact that these algorithms form a family of training procedures that yield similar functional representations when applied to a class of parallel distributed networks, and that they can do so using mechanisms not excluded, and perhaps suggested, by neurophysiological evidence.

### References

- [1] Andersen RA. Visual and eye movement functions of the posterior parietal cortex. *Ann Rev Neurosci*, 12:377-403, 1989.
- [2] Andersen RA, Zipser D. The role of the posterior parietal cortex in coordinate transformations for visual-motor integration. *Can J Physiol Pharm*, 66(4):488-501, 1988.
- [3] Barto AG. Learning by statistical cooperation of self-interested neuron-like computing elements. *Human Neurobiol*, 4:229-256, 1985.
- [4] Barto AG. From chemotaxis to cooperativity: abstract exercises in neuronal learning strategies. In: Durbin RM, Miall RC, Mitchison GJ, eds. *The computing neuron*. New York: Addison-Wesley, 1989.
- [5] Barto AG, Anandan P. Pattern recognizing stochastic learning automata. *IEEE Trans on Systems, Man, and Cybern*, 15:360-375, 1985.
- [6] Barto AG, Jordan MI. Gradient following without backpropagation in layered networks. *Proc IEEE First Ann Intern Conf on Neural Networks*, II:629-636, San Diego, CA, June 1987.
- [7] Hecht-Nielsen R. Theory of the backpropagation neural network. *Intern Joint Conf on Neural Networks*, I:593-605, Washington, D.C., June 1989.
- [8] Hinton GE. Connectionist learning procedures. *Tech. Rep. No. CMU-CS-87-115*. Pittsburgh, PA: Carnegie-Mellon University, 1987.
- [9] Linsker R. How to generate ordered maps by maximizing the mutual information between input and output signals. *IBM Tech. Rep. No. 65530*. Yorktown Heights, NY: Watson Research Center, 1989.
- [10] Mazzoni P, Andersen RA, Jordan MI. A more biologically plausible learning rule applied to a network model of cortical area 7a. *J Neurosci*, submitted.
- [11] Parker DB. Learning-logic. *TR-47*. Cambridge, MA: Center for Computational Research in Economics and Management Science, 1985.
- [12] Richardson RT, Mitchell SJ, Baker FH, DeLong MR. Responses of nucleus basalis of Meynert neurons in behaving monkeys. In: Woody CD, Alkon DL, McGaugh JL, eds. *Cellular mechanisms of conditioning and behavioral plasticity*. New York: Plenum Publ. Corp., 1988.
- [13] Rumelhart DE, McClelland JL, PDP Research Group. *Parallel distributed processing*. Cambridge, MA: MIT Press, 1986.
- [14] Sejnowski TJ. Skeleton filters in the brain. In: Hinton GE, Anderson JA, eds. *Parallel models of associative memory*. Hillsdale, N.J.: Lawrence Erlbaum Associates, 1981.
- [15] Sejnowski TJ, Chattarji S, Stanton PK. Introduction of synaptic plasticity by Hebbian covariance in the hippocampus. In: Durbin RM, Miall R C, Mitchison GJ, eds. *The computing neuron*. New York: Addison-Wesley, 1989.
- [16] Williams RJ. A class of gradient-estimating algorithms for reinforcement learning in neural networks. *Proc IEEE First Ann Intern Conf on Neural Networks*, II:601-608, San Diego, CA, June 1987.
- [17] Zipser D, Andersen RA. A backpropagation programmed network that simulates response properties of a subset of posterior parietal neurons. *Nature*, 331:679-684, 1988.