to a 6-hydroxydopamine lesion of the substantia nigra, Part 1 Pharmacol 58:99-100.
Sommerling JL, Macgregor Leo P, Curran T, Morgan JI (1988) Dynamic alterations occur in the level and co- 
Strecker HE, Jacobs B (1985) Substance NG dopamine- 

A More Biologically Plausible Learning Rule Than Backpropagation Applied to a Network Model of Cortical Area 7a

Area 7a of the posterior parietal cortex of the primate brain is concerned with represented head-centered space by combining information about the retinal location of a visual stimulus and the position of the eyes in the orbits. An artificial neural network was previously trained to perform this coordinate transformation task using the backpropagation learning procedure, and units in its middle layer (the hidden units) that display properties very similar to those of area 7a neurons presumed to code for spatial location (Andersen and Zipser, 1988; Zipser and Andersen, 1988). We developed two neural networks with architecture similar to Zipser and Andersen’s model and trained them to perform the same task using a more biologically plausible learning procedure than backpropagation. This procedure is a modification of the Associative Reward-Penalty (ARP) algorithm (Baro and Anandan, 1985), which adjusts connection strengths using a global reinforcement signal and local synaptic information. Our networks learn to perform the tasks successfully to any degree of accuracy and almost as quickly as with backpropagation, and the hidden units develop response properties very similar to those of area 7a neurons. In particular, the probability of firing of the hidden units in our networks varies with eye position in a roughly planar fashion, and their visual receptive fields are large and have complex surfaces. The synaptic strengths computed by the ARP algorithm are equivalent to and interchangeable with those computed by backpropagation. Our networks also perform the correct transformation on pairs of eye and retinal positions never encountered before. All of these findings are unaffected by the interposition of an extra layer of units between the hidden and output layers. These results show that the response properties of the hidden units of a layered network trained to perform coordinate transformations, and their similarity with those of area 7a neurons, are not a specific result of backpropagation training. The fact that they were obtained by a more biologically plausible learning rule corroborates the validity of this neural network’s computational algorithm as a plausible model of how area 7a may perform coordinate transformations.

An important element of information processing in the nervous system appears to be the collective behavior of large ensembles of neurons. The study of the emergent properties of these networks has been an important motivation behind the development of artificial neural network models whose architecture is inspired by the biological wiring of nervous systems, containing a large number of computationally extensive units connected to one another. It is the hope of many neuroscientists that these models will elucidate, at least at an abstract level, some of the basic principles involved in information handling by the nervous system and thus perhaps provide a theoretical framework within which to formulate experimental questions. One of the best examples of this type of approach so far is a neural network model of area 7a of the primate’s posterior parietal cortex developed by Zipser and Andersen (1988; Andersen and Zipser, 1988). From lesion and single-unit recording estimates, it appears that area 7a is concerned with the representation of spatial locations in a head-centered reference frame (for a review, see Andersen, 1989). This representation is distributed over a group of neurons that are sensitive to both the position of the eyes in the orbits and the location of visual stimuli on the retinas. Other neurons in area 7a respond to either eye position or visual stimuli alone and are presumed to provide inputs from which different eye position receptors extract the cranioptatic representation. The latter neurons have very large retinotopic visual receptive fields and their response to eye position interacts nonlinearly with the visual signals. Although the majority of area 7a neurons maintain the same retinotopic receptive fields for different eye positions, the magnitude of the visual response varies with angle of gaze. Holding the retinal location of a visual stimulus constant and varying angle of gaze, Andersen and his colleagues found that these neurons’ overall firing rate (visual plus eye position component) varied roughly linearly with changes in hori zontal and vertical eye position (Fig. 1b). The re sponse profiles for varying eye position were called “gain fields,” and a majority (50%) of area 7a cells had planar or largely planar gain fields (Fig. 1c; Andre nsen et al., 1985; Andersen and Zipser, 1988; Zipser and Andersen, 1988). Zipser and Andersen (1988) designed a computer

Pietro Mazzoni,1,2 Richard A. Andersen,3 and Michael J.ordan 1Department of Brain and Cognitive Sciences and 2Harvard-MIT Division of Health Sciences and Technology, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

Centrally funded by grants from the National Institutes of Health.
simulated neural network with an input layer, a layer of internal or hidden units, and an output layer. The input layer consisted of two groups of units with properties similar to those of area 7a neurons sensitive to either eye position or visual stimulus alone. The input layer coded for head-centered location in an abstract format independent of eye position and was used to generate error signals to train the network. The network was trained to perform the coordinate transformation from retinotopic to craniotopic reference frames using the backpropagation procedure (Rumelhart et al., 1986a). The striking result of this study was that in the process of learning the hidden units developed response properties very similar to those of the area 7a neurons that seem to encode spatial location—specifically, a roughly planar modulation of visual response by eye position, and a single unit in the next layers of the network. This suggests that area 7a neurons, as an ensemble, can in fact provide information for the abstract representation of space, and that these neurons’ properties can be generated by a supervised learning paradigm.

Backpropagation networks can learn to perform a computation without explicit “knowledge,” using only error signals from the environment as cues to improve its performance, in a paradigm referred to as “supervised learning” (Hinton, 1987). This type of training schemes has converged upon neural networks a strong element of biological plausibility than many previous models of brain function that relied on precompiled rules and symbol processing. Moreover, although various supervised learning algorithms for one-layer networks were described long before backpropagation (Widrow and Hoff, 1960; Minsky and Papert, 1969), the discovery of the backpropagation algorithm (Werbos, 1974; Parker, 1985; Rumelhart et al., 1986a), which can be applied to more powerful multi-layer networks composed of nonlinear units, is in large part responsible for the recent increase in interest in neural network models. Despite the biological plausibility of supervised learning, however, the implementation of backpropagation in the nervous system would require mechanisms such as the retrograde propagation of signals along axons and through synapses and precise error signals that are different for each neuron, which are not accepted as likely candidates for learning processes in the brain. To this end, Zipser and Anderson (1988) emphasized that it was the solution that was of interest in their model and not the method by which this solution was learned. They speculated that other, more biological learning procedures might generate a solution to the coordinate transformation task similar to that which resulted from backpropagation learning. It was therefore important to ask how crucial backpropagation is for the development of the hidden units’ properties in a model like Zipser and Anderson’s.

We addressed precisely this question in our study. We trained two neural networks with architectures similar to the Zipser and Anderson model using a supervised learning paradigm that is more plausible from a biological perspective than backpropagation. The algorithms we used, which are a variant of the Associative Reward–Penalty (ARP) algorithm for supervised learning tasks introduced by Barto and Jordan (1987), train a neural network using a global reinforcement signal broadcast to all the connections in the network. We found that our networks can indeed be trained by these algorithms to perform the coordinate transformation task, and that the hidden units acquire response properties very similar to those of area 7a neurons, as in the unsupervised model. A second layer of hidden units can be interposed between the original hidden layer and the output layer without altering the properties developed by units in the first hidden layer. Furthermore, all of our networks perform the correct transformation on pairs of eye and retinal positions that were not experienced before that is, they generalize appropriately. A less detailed report of results from one of the ARP networks has recently been published (Mazzoni et al., 1991).

Materials and Methods

Model Networks

We devised two types of networks that we trained to map visual targets to head-centered coordinates, giving any arbitrary pair of eye and retinal positions. The basic architecture of these networks is similar to that of Zipser and Anderson’s model.

Mixed ARP Network

We call the first network the Mixed ARP network (Fig. 2A) because its hidden and output layers are trained by different learning rules (see Training, below). It is composed of three layers of computing units: an input, a hidden, and an output layer. The network has a fully connected feedforward architecture, meaning that every unit in each layer sends a signal to every unit in the next layer, with an individual connection strength or weight (w), so that signals propagate in one direction from the input toward the output layer. The input layer consists of two groups of units (Fig. 2A, squares), one coding for the retinal location of the visual stimulus, and the other for the position of the eyes in the orbits. These units encode the external visual stimulus and transforming an angular position into a value between 0 and 1, which is then sent to the hidden units. Retinotopic locations are represented by 64 visual units arranged in an 8 × 8 array, each with a gaussian receptive field (Fig. 2B) with peak at 10° from its neighbors and 1/30° width of 15°, producing a uniform topographic representation of the retina. Eye position is coded by four sets of eight units representing horizontal and vertical eye coordinates with positive and negative signs, for which activation is a linear function of eye angle (Fig. 2C). These representations were modeled according to characteristics of area 7a neurons established in previous studies (Andersen et al., 1985; Andersen and Zipser, 1988) and are the same...
simulated neural network with an input layer, a layer of internal or hidden units, and an output layer. The input layer consisted of two groups of units with properties similar to those of area 7a neurons sensitive to either eye position or visual stimulus alone. The output layer coded for head-centered location in an abstract format independent of eye position and was used to generate error signals to train the network. The network was trained to perform the coordinate transformation from retinotopic to craniotopic reference frames using the backpropagation procedure (Rumelhart et al., 1986a). The striking results of this study were that in the process of learning the hidden units developed response properties very similar to those of the area 7a neurons that seem to encode spatial location—specifically, a roughly planar modulation of visual response by eye position, and a unit in the next layer which suggested that area 7a neurons, as an ensemble, can in fact provide information for the abstract representation of space, and that these neurons’ properties can be generated by a supervised learning paradigm.

Backpropagation networks can learn to perform a computation without explicit “knowledge,” using only error signals from the environment as cues to improve its performance, in a paradigm referred to as “supervised learning” (Hinton, 1987). This type of training scheme has been applied to neural networks a structure element of biological plausibility than many previous models of brain function that relied on precomputed rules and symbol processing. Moreover, although various supervised learning algorithms for one-layer networks were described long before backpropagation (Widrow and Hoff, 1960; Minsky and Papert, 1969), the discovery of the backpropagation algorithm (Werbos, 1974; Parker, 1985; Rumelhart et al., 1986a), which can be applied to more powerful multilayer networks composed of nonlinear units, is a large part responsible for the recent increase in interest in neural network models. Despite the biological plausibility of supervised learning, however, the implementation of backpropagation in the nervous system would require mechanisms such as the retrograde propagation of signals along axons and through synapses and precise error signals that are different for each neuron, which are not accepted as likely candidates for learning processes in the brain. To address this issue, Zipser and Andersen (1988) emphasized that it was the solution that was of interest in their model and not the method by which this solution was learned. They speculated that, in another example, more biological learning procedures might generate a solution to the coordinate transformation task similar to that which resulted from backpropagation learning. It was therefore important to ask how crucial backpropagation is for the development of the hidden units’ properties in a model like Zipser and Andersen’s.

We addressed precisely this question in our study. We trained two neural networks with architectures similar to the Zipser and Andersen model using a supervised learning paradigm that is more plausible from a biological perspective than backpropagation. The algorithm used was that of the associative Hebbian–Penalty (AHP) algorithm for supervised learning tasks introduced by Barto and Jordan (1987). We trained a neural network using a global reinforcement signal broadcast to all connections in the network. We found that our networks could indeed be trained by these algorithms to perform the coordinate transformation task, and that the hidden units acquire response properties very similar to those of area 7a neurons, as in the classical model. A second layer of hidden units can be interposed between the original hidden layer and the output layer without altering the properties developed by units in the first hidden layer. Furthermore, all of our networks perform the correct transformation on pairs of eye and retinal positions that were not presented before that is, they generalize appropriately. A less detailed report of results from one of the AHP networks has recently been published (Mazzoni et al., 1991).

Materials and Methods

Model Networks

We devised two types of networks that we trained to map visual targets to head-centered coordinates, given an arbitrary pair of eye and retinal positions. The basic architecture of these networks is similar to that of Zipser and Andersen’s model.

Mixed αH Network

We call the first network the Mixed αH network (Fig. 2a) because its hidden and output layers are trained by different learning rules (see Training, below). It is composed of three layers of computing units: an input, a hidden, and an output layer. The network has a fully connected feedforward architecture, meaning that every unit in each layer sends a signal to every unit in the next layer. We can derive an individual connection strength or weight (w), so that signals propagate in one direction from the input toward the output layer. The input layer consists of two groups of units (Fig. 2a, squares), one coding for the retinal location of the visual stimulus, and the other for the position of the eyes in the orbit. These units encode the external visual stimuli and transform an angular position into a value between 0 and 1, which is then sent to the hidden units. Retinotopic locations are represented by 64 visual units arranged in an 8 × 8 array, each with a gaussian receptive field (Fig. 2b) with peak at 0° from its neighbors’ 1/8 width of 15°, producing a uniform topographic representation of the retina. Eye position is coded by four sets of eight units representing horizontal and vertical eye coordinates with positive and negative slopes, for which activation is a linear function of eye angle (Fig. 2c). These representation forms were modeled according to characteristics of area 7a neurons established in previous studies (Andersen et al., 1985, 1988; Zipser and Andersen, 1988) and are the same
The hidden units project in turn to the output layer, which encodes the craniotopic location that is the vector sum of the positions encoded by the retinal and eye position inputs. The units in the output layer (Fig. 2a, circles) are deterministic logistic elements (Fig. 3b). Like the binary stochastic units, they too perform a weighted sum of their inputs and pass it through the logistic function. In the deterministic logistic element, however, the value between 0 and 1 is the unit's output itself. The output, therefore, is continuous and precisely predictable from the input. In the analogy with the neuron, this continuous output would correspond to the firing rate. The outputs of the units in the output layer encode head-centered location in one of two possible formats: a "monotonic" representation analogous to eye position input, containing any number of units from 2 to 52 (Fig. 2a, output 1), and a "gaussian" representation similar to that of the retinal input, with a number of units ranging from 4 to 64 (Fig. 2a, output 2). In the monotonic representation, the activity of the output units in the output layer is for visual stimuli appearing within the limits of the foveal representation. A continuous output is associated with a value σ in the range of 0 to 1. The output units produce the foveal function for the ALA network. A Gaussian monotonometric function could be a motor signal to the extrinsic eye muscles (Zipser and Andersen, 1989). Generalization of the foveal function is important for more than one receptive field for a mental representation of craniotopic space. These output representations are similar to those of the Zipser and Andersen model.

**All A Network**

The second type of network we studied was the All A network (Fig. 4a), so called because all of its connections are adjusted by the All A algorithm (see Training, below). This network's input and hidden units are identical to those of the Mixed A network. The output layer, however, is composed of binary stochastic units like the hidden layer. Its control of the craniotopic location in one of two alternative formats. Due to the binary nature of the output units, we devised output format for the All A network such that the collective activity of the output units codes for discrete adjacent regions of space, instead of continuous varying spatial locations. In the "binary monotonic" format, four triplets of units divide craniotopic space into 16 regions by giving an output of 1 if the x (or y) head center is within the region of interest. Then, the hidden units between the first layer of hidden units and the output layer. This extra layer, like the original hidden layer, is also composed of binary stochastic units. We did this to see that the output is the region of the output units in the hidden layer of the three layer networks depended directly on a direct connection with the output layer.

For comparison purposes, we also set up a back-propagation network identical to the Mixed A network described in Figure 2a, except for its hidden units, which are deterministic logistic elements and not binary stochastic elements.

**Training**

We trained our networks to perform the coordinate transformation task in a supervised learning paradigm. In supervised learning, the network starts out with all connections and biases set to 0, or at some set of small random values if the training algorithm cannot break the initial symmetry (the All A algorithm). The network is first trained with a 1000 training inputs, each with a 0.5 probability of being a 0 and a 0.5 probability of being a 1, and each training input corresponds to a random location in the output space.

Each training input is presented to the network's input layer, which propagates a signal to the following layers (Fig. 5). The output layer produces a "quiescent" output based on the initial set of connections. This output is compared to the correct output pattern for that particular input, and the error is computed and fed back to the network (Fig. 5, broken arrows). The values of all the network's weights and biases are then adjusted by a learning rule designed to minimize the error. This process is repeated until the error is reduced to a value below a desired level.

For our task, the input pattern is a signal for the retinal location of a target in an environment with one position corresponded to the desired position. The desired output pattern is one that codes for the head-centered location...
as those used in the input layer of the Zipser and Andersen model.

The hidden layer (Fig. 2a, diamonds) is so described because it is not "visible" (i.e., directly connected to external agents acting at the input or at the output. The type of computational unit making up this layer is the binary stochastic element (Fig. 1b). The probabilistic element making up the input is a weighted sum (s) of its inputs and passes it through the logistic function to obtain a value (p) between 0 and 1. This value is then used as the probability of producing an output equal to 1. The output is 0 with probability 1 - p. This type of computing element is "neurally inspired" in the sense that it can occur in some well-established features of neurons. In such an analog, the inputs correspond to synaptic inputs from other neurons, the connection weights represent synaptic strength (with inhibitory synapses implemented as negative weights), and the weighted input represents the intracellular potential. The probability of firing is analogous to a neuron's rate of action potential firing, and changes in the total weighted input affect the neuron's firing rate in a manner similar to changes in the intracellular potential affecting a neuron's firing rate. This hidden layer differs from that of the Zipser and Andersen network in that the units of the latter were deterministic logical elements (discussed below). The number of hidden units in the Mixed_A network, as well as in all the networks described to now, varied from two to twelve.

The hidden units project in turn to the output layer, which encodes the craniotopic location that is the vector sum of the positions encoded by the retinal and eye position inputs. The units in the output layer (Fig. 2a, circles) are deterministic logical elements (Fig. 3b). Like the binary stochastic units, they too perform a weighted sum of their inputs and pass it through the logistic function. In the deterministic logical element, however, the value between 0 and 1 is the unit's output itself. The output, therefore, is continuous and precisely predictable from the input. In the analogy with the neuron, this continuous output would correspond to the firing rate. The number of the output units encode head-centered location in one of two possible formats: a "monotonic" representation analogous to eye position input, containing any number of units from 2 to 52 (Fig. 2a, output 1), and a "gazaine" representation similar to that of the retinal input, with a number of units ranging from 4 to 64 (Fig. 2a, output 2). In the monotonic representation, the activity of the output units in a row for more peripheral locations of the visual target with respect to the head, regardless of eye position. The gaussian format units fire for visual stimuli appearing within a distance of the head-centered coordinates. We used this representation in our model, as well as in the output layer of the Mixed_A network. A (binary) logical element in the two-dimensional representation could be a motor signal to the extrinsic eye muscles (Zipser and Andersen, 1988; Goodman and Andersen, 1990), whereas the gaussian format units are more like a receptive field for a mental representation of craniotopic space. These output representations are similar to those of the Zipser and Andersen model.

All_A network

The second type of network we studied is the All_A network (Fig. 4a), so called because all of its connections are adjusted by the A_up algorithm (see Training, below). This network's input and hidden layers are identical to those of the Mixed_A network. The output layer, however, is composed of binary stochastic units like the hidden layer. This, too, encodes the craniotopic location in one of two alternative formats. Due to the binary nature of the output units, we defined output format for the All_A network such that the collective activity of the output units codes for discrete adjacent regions of space, instead of continuously varying spatial locations. In the "binary monotonous" format, four triplets of units divide craniotopic space into 16 regions by giving an output of 1 if the (x, y) head-centered coordinate is within the region (Fig. 4b). This format is analogous to the monotonic format of the Mixed_A network. The binary counting format with an extra layer of hidden units is the "binary gaussian" format, in which four units have overlapping receptive fields centered at (±0.60, ±0.60), such that each unit's output is 1 if the spatial position is within 100° of its center (Fig. 4c). This format divides craniotopic space into 16 regions by virtue of the overlap of the receptive fields. The number of units in both types of binary output format may be increased to improve the output's spatial resolution. We did not examine the parameter of number of output units systematically, as it did not produce significantly different network behavior.

Other Networks

In addition to the twoingo-three-layer networks just described, we studied the behavior of two similar four-layer networks. These consist of a Mixed_A network and an All_A network, stacked with an extra layer of hidden units between the first layer of hidden units and the output layer. This extra layer, like the original hidden layer, is also composed of binary stochastic units. We did this to see whether the response properties developed by the units in the hidden layer of the three-layer networks depended on a direct connection with the output layer. For comparison purposes, we also set up a back-propagation network identical to the Mixed_A network described in Fig. 2a, except for its hidden units, which are deterministic logical elements and not binary stochastic elements.

Training

We trained our networks to perform the coordinate transformation task in a supervised learning paradigm. In supervised learning, the network learns out with all connections and biases set at 0, or at some set of small random values if the training algorithm cannot break the initial symmetry (the A_up algorithm we used can break both cases). An input pattern is presented to the network's input layer, which propagates a signal to the following layers (Fig. 5, solid arrows). The output layer's outputs produce a "guessed" output based on the initial set of connections. This output is compared to the correct output pattern for that particular input, and, if it is not correct, is fed back to the network (Fig. 5, broken arrows). The values of all the network's weights and biases are then adjusted by a learning procedure that at the next generation of the same pattern the error is, at least on average, smaller than before. This procedure is repeated until the error is reduced to a value below a desired level.

For our task, the input pattern is a signal for the retinal location of a stimulus paired with one for the current eye position. The desired output pattern is one that codes for a head-centered location of
that is the vector sum of the retinal and eye positions. The error signal is computed externally to the network to the network.

To draw 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 make a movement toward the stimulus, and any metric of performance, such as visually detected inaccuracies, could be used to generate an error signal.

The network is trained by being repeatedly presented with a finite number of patterns forming a chosen training set, the connection weights being adjusted after each pattern presentation. We used two types of pattern sets to train the networks. One is a set of randomly chosen horizontal and vertical locations and eye positions so that the desired output associated with each input is an arbitrary location in head-centered space. In the analogy with the learning animal, learning with this set would correspond to looking at various stimuli in the visual field at various angles of gaze. The other type of training set consists of input patterns for a given eye position randomly chosen, so that the eye position is randomly chosen, while the retinal location is computed so that the vector sum of the two inputs is one of a few chosen craniotopic locations. The resulting training set contains a few fixed spatial locations, each represented by a large number (at least 10) of retinal and eye-position neurons that add vectorially to it. For an animal, this type of training corresponds to looking at an object fixed in space with the eye in various orientational orientations. The training set used was used to see how well the network generalized to new locations in space once it had been trained on a few examples.

We devised two variants of the supervised learning procedure for \( a_{n} \) networks of Barbo and Jordan (1987) to adapt to changes in the visual input. For the supervised learning paradigm, depends on how close the current output is to the desired output. Specifical y, \( r_{n} \) is a value between 0 and 1, indicating maximum error in the output (i.e., every unit that should be firing is not, and vice versa), 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 so as to minimize the value of this error.

The equation for updating the weights on a binary stochastic unit is

\[
\Delta w_{i} = r_{n} (x_{i} - x_{n}) f'(a_{n}) x_{n} \]

where \( x_{n} \) indicates the desired output of the \( n \)th unit, \( x_{n} \) is a scalar learning rate, \( f'(a_{n}) \) is the derivative of the logistic function with respect to the unit's net input \( a_{n} \), and \( x_{n} \) is the output of one of the units' "prey" to "eat" in the output layer. Typical values for \( r_{n} \) vary from 0.2 to 0.4, with a range between 0.1 and 0.3. The learning rate \( r_{n} \) is inversely proportional to the number of units in the hidden layer, with a constant learning rate used for all units. The logistic function, which is a sigmoidal function, is used because it is differentiable and has a smooth, continuous derivative, allowing for efficient computation of the error signal.

Results

Learning All the networks described above learned to perform the coordinate transformation as accurately as possible. Figure 6 shows the general behavior during training of the two \( \text{At}_{n} \) networks studied and compares them to that of a backpropagation network, with identical architecture, learning from the same training set. We plot here for clarity the value of the network's error, averaged over the number of output units and the number of patterns in the training set, versus the number of iterations of the complete training set. The \( \text{At}_{n} \) networks produce learning curves with much more jitter than the curve for backpropagation training, but maintain the same nature of their hidden units and to the type of error signal used in \( \text{At}_{n} \) training (see Discussion). All three networks, however, produce curves with similar shapes, and the times required for convergence are comparable. For the backpropagation network, which has a continuous output, the error decreases monotonically (Fig. 6a), while for the All \( \text{At}_{n} \) network, which has a binary output, the error follows a noisy path down to 0 and spends increasing more time there, flickering occasionally to the value of the output's smallest resolvable value (Fig. 6b). The error for the Mixed \( \text{At}_{n} \) network is also noisy, because this network's hidden units are stochastic. It assumes, however, a continuous range of values (Fig. 6b), because the output units are logistic classifiers of the training curves for which were obtained for both monotonic and gaussian formats. Neither algorithm had serious problems with local minima (i.e., getting stuck at suboptimal solutions).

Response Properties We examine some of the properties developed by the hidden units during training in the same manner as Zipser and Anderson did for their model, except we plot the frequency responses of our hidden units. We plot the frequency response of the hidden units for the change in the frequency with which a binary unit's output is 1 encodes a continuous value, which can be conceived as a firing rate.

An interesting feature of area 7a neurons is that the visual and the eye position contributions to their overall response interact nonlinearly. For a constant retinal stimulus position, the total response is not composed of a constant visual response to which an independently varying amount of activity is added as the eye position changes. As Figure 7a and the work of Anderson and Zipser (1988) and Zipser and Anderson, 1990b shows, the response of area 7a neurons can vary simultaneously, in either the same or opposite directions, or to different degrees with eye position. The overall response of the hidden units, moreover (Fig. 7, thin circles), was always roughly planar along vertical and horizontal eye positions. This result was found in 78% of spatially tuned area 7a neurons (Anderson and Zipser, 1988). 2
that is the vector sum of the retinal and eye positions.

The error signal is computed externally to the network.

To draw 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 proprio-
cceptive or corollary discharge pathways). The animal
may then make a movement toward the stimulus, and
any metric of performance, such as visually detected
inaccuracies, could be used to generate an error sig-
nal.

The network is trained by being repeatedly pre-
sented with a finite number of patterns forming a
chosen training set, the connection weights being
adjusted after each pattern presentation. We used two
types of pattern sets to train the networks. One is a set
of 200 random patterns, each consisting of K retinal
locations and eye po-
sitions so that the desired output associated with
each input is an arbitrary location in head centered space.
In the analogy with the learning animal, learning with
this set would correspond to looking at various stimuli
in the visual field at various angles of gaze. The other
type of training set consists of input patterns for which
the eye position is randomly chosen, while the retinal
location is computed so that the vector sum of the
two inputs is one of a few chosen cranialopic loca-
tions. The resulting training set contains a few fixed
spatial locations, each represented by a large number
at least 10) of retinal and eye positions that add
vectorially to it. For an animal, this type of training
corresponds to looking at an object fixed in space
with the eyes in various coronal orientations. The
training set was used to see how well the network
generalized to new locations in space once it had been trained on a few few.

We devised two variants of the supervised learning
procedure for $\mathcal{A}_n$ networks of Barto and Jordan
(1987) to approximate the two types of our training
strategy, which is the $\mathcal{A}_n$ learning rule. Every binary
stochastic element in a given network receives a scalar
reinforcement (or signal) $s$ whose value, in the
supervised learning paradigm, depends on how
close the current output is to the desired output.
Specif-
ically, $s$ is a value between 0 and 1, typically
maximum error in the output (i.e., every unit
that should be firing is not, and vice versa), 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 in the same way as to
minimize the value of this payoff. Using the notation
of Figure 3a, where $x_i$ represents the output of the
ith unit in the network, $p_{x_i}$ is probability of firing, and
$w_{ij}$ the connection weight for its input from the
jth unit, the equation for updating the weights on a
binary stochastic unit is

$$
\Delta w_{ij} = p_{x_i}(1-x_i)  f(x_j) - x_i f(x_j)
$$

where $x_i$ indicates the $i$th output unit, $x_i$ the
desired output of the $i$th unit, $a$ is a scalar learning rate,
and $f$ is the derivative of the logistic function with respect
to the unit's net input $s_i$ and $x_i$ is the output of one of
the hidden units "preprocessing" to it. Typical values
for $\alpha$ are between 0 and 1, and for $\gamma$ between 0 and 1, and
can vary simultaneously, in either the same or op-
opposite directions, or to different degrees with eye
positions (see Anderson and Zipser, 1988, for a more
detailed analysis of area 7a gain fields). When ex-
amined after training, the hidden units of both types
of networks displayed gain fields similar to those of
area 7a neurons (Fig. 7b), as well as the same type
of variation. The overall response of the hidden
units, moreover (Fig. 7, thin circles), was always
depicted very similar to the spatially tuned gain fields
resulting from the equation $\Delta y = \gamma \Delta x$, or
identical architecture, learning from the same
training set. We plot the absolute value of the output
units' error, averaged over the number of output units
and the number of patterns in the training set, versus the
number of iterations of the complete training
set. The $\mathcal{A}_n$ network produces learning curves with
more noise than the curve for backpropagation
training due to the nature of their hidden
units and to the type of error signal used in $\mathcal{A}_n$ train-
ing (see Discussion). All three networks, however,
produce curves with similar envelopes, and the times
required for convergence are comparable. For the
backpropagation network, which has a continuous
output, the error decreases monotonically (Fig. 6a),
while for the All $\mathcal{A}_n$ network, which has a binary
output, the error follows a noisy path down to 0 and
spends increasingly more time there, flickering oc-
casionally to the value of the output's smallest res-
solvable angle (Fig. 6c). The error for the Mixed $\mathcal{A}_n$
network is also noisy, because this network's hidden
units are stochastic. It assumes, however, a con-
tinuous range of values (Fig. 6b), because the output units are
generative elements, and the training curves for which were
obtained for both monotonic and gaussian formats.
Neither algorithm had serious problems with local
minima (i.e., getting stuck at suboptimal solutions).

Response Properties
We investigated some of these properties developed by the
hidden units during training in the same manner as
Zipser and Anderson did for their model, except we plot
the error with respect to the firing rate and thus
continuous variable) instead of its instantaneous output
(which is binary). The probability of firing can be thought of as equivalent to the firing rate and thus
continuous to the equivalent output in the Zipser and
Anderson model. In other words, over a number of repeated presentations of the same input, the
frequency with which a binary unit's output is 1 en-
codes a continuous value, which can be conceived as a
driving force for learning.

An interesting feature of area 7a neurons is that
the visual and the eye position contributions to their over-
all response interact nonlinearly. For a constant re-
Fential stimulus position, the total response is not com-
pounded of a constant visual response to which an
independently varying amount of activity is added as
to the eye position contribution. As shown in Figure 7a and the
work of Anderson and Zipser (1988). Zipser and Anderson,
1988, demonstrated that a model with stochastic eye and
eye position components also is the basis for the backpropagation algorithm and is used in identical
form to train the output units of
backpropagation networks.

Results

Learning

All the networks described above learned to perform
the coordinate transformation task. Figure 6 shows the
general behavior during training of the two $\mathcal{A}_n$
networks studied and compares them to that of a backpropagation network, with
a second hidden layer of binary stochastic units was added to either the Mixed $\mathcal{A}_n$ or the All $\mathcal{A}_n$ network, both networks learned to perform the task, and the units in the first hidden layer still developed planar
gain fields similar to those of area 7a (Fig. 7a only
Apart from that, it is worth noting that when studied in more detail, that is, when sampled over a larger range of eye
positions, the gain fields produced by $\mathcal{A}_n$ (as well as backpropagation) training are not exactly planar, but
roughly sigmoidal along one direction of eye position (Fig. 8). In other words, the overall responses are approximately planar within a range of eye positions and are flat outside this range. It turns out that this range is determined by the most eccentric eye positions on which the network was trained. The unit whose gain field is shown in Figure 8, for example, was part of a network trained with eye positions between −60° and 60° (horizontally as well as vertically), and it developed a gain field that is planar along the y direction. This range shows that the hidden units learn to interpolate for eye positions between those in the training set, but they do not learn to extrapolate to eye positions outside this range. We believe that this is a direct consequence of using a sigmoidal probability function (or output function in the case of the deterministic logistic element) for the hidden units. The gain fields of area 7a neurons may also flatten outside a certain range of eye positions, producing a sigmoidal shape like that in Figure 8. At present, the recording data available are too noisy, and perhaps too limited in range of eye positions, to distinguish between a simple plane and a sigmoid for the gain fields.

There was also a qualitative similarity between the receptive fields developed by the network's hidden units and those of area 7a neurons, as shown in Figure 9. The most striking feature of these neurons' receptive fields is their size, which extends to diameters of 80° (Fig. 9a). This feature is reproduced by our model networks (Fig. 9c). Another feature is the complexity of these receptive fields' surfaces, characterized by one or more smooth peaks of various eccentricities, which sets area 7a neurons apart from those of many other visual areas. The networks' hidden units also display a similar complexity in their receptive fields, although such a comparison can be qualitative at best. As was the case for the gain fields, the addition of an extra hidden layer did not significantly affect the types of receptive fields developed by units in the first hidden layer (Fig. 9d) only All Aa shown.

The response properties of the Aa networks' hidden units are similar not only to those of area 7a neurons, but also to those of hidden units of networks trained by backpropagation to compute coordinate transformations. These response properties were described by Zipser and Anderson (1988) and Zipser and Andersen, 1988. We were able to reproduce them and we can backpropagation networks with the same number of units and the same training set as the Aa networks (Fig. 10). This similarity suggests that the S model Aa rule and backpropagation compare similar solutions (i.e., sets of network connection strengths) to the coordinate transformation problem.

**Comparison of Solutions**

The solutions found by S model Aa training and by backpropagation are not just similar in the qualitative sense described in Figures 7, 9, and 10. In fact, for a given training set, we found that the set of weights trained by the Aa algorithm may be transferred to a backpropagation network (with continuous output units) without any appreciable reduction in the accuracy of the network's response to that training pattern, and vice versa (Fig. 11). This is true for both versions of our networks (Mixed and All Aa) and for networks with one and two hidden layers, as long as the output format is the same for the Aa and backpropagation networks being compared. The individual weights of the weights are not the same after training by the different procedures, but the overall structure of these weights is such that the two algorithms' solutions to the coordinate transformation problem are functionally equivalent for the various network structures.

**Generalizations**

A property that is often exhibited by artificial neural networks trained by examples is the ability to generalize from those examples, that is, to produce the correct output when presented with input patterns that were not in the training set. This property is of great theoretical and practical importance, as it dem- onstrates that the model network has not merely learned to associate patterns in the training set with their correct outputs on an individual basis, but has
The receptive fields of area 7a neurons are roughly sigmoidal along one direction of eye position (Fig. 8). In other words, the overall responses are approximately planar within a range of eye positions and are flat outside this range. It turns out that this range is determined by the most eccentric eye positions on which the network was trained. The unit whose gain field is shown in Figure 8, for example, was part of a network trained with eye positions between -40° and 40° (horizontally as well as vertically), and it developed a gain field approximately planar over this range along the x direction (there were other units in the network with similar gain fields oriented along the y direction). This result shows that the hidden units learn to interpolate for eye positions between those in the training set, but they do not learn to extrapolate to eye positions outside this range. We believe that this is a direct consequence of using a sigmoidal probability function (or input-output function) in the case of the deterministic logistic element) for the hidden units. The gain fields of area 7a neurons may also flatten outside a certain range of eye positions, producing a sigmoidal shape like that in Figure 8. At present, the recording data available are too noisy, and perhaps too limited in range of eye positions, to distinguish between a simple plane and a sigmoid for the gain fields.

There was also a qualitative similarity between the receptive fields developed by the network’s hidden units and those of area 7a neurons, as shown in Figure 9. The most striking feature of these neurons’ receptive fields is their size, which extends to diameters of 80° (Fig. 9a). This feature is reproduced by our model networks (Fig. 9b,c). Another feature is the complexity of these receptive fields’ surfaces, characterized by one or more smooth peaks of various eccentricities, which sets area 7a neurons apart from those of many other visual areas. The networks’ hidden units also display a similar complexity in their receptive fields, although such a comparison can be qualitative at best. As was the case for the gain fields, the addition of an extra hidden layer did not significantly affect the types of receptive fields developed by units in the first hidden layer (Fig. 9d; only All-A shown).

The response properties of the All-A networks’ hidden units are similar not only to those of area 7a neurons, but also to those of hidden units of networks trained by backpropagation to compute coordinate transformations. These response properties were described by Zipser and Andersen (1988) and Zipser (1988). We were able to reproduce them also in a backpropagation network with the same number of units and the same training set as the All-A networks (Fig. 10). This similarity suggests that the S model All-A rule and backpropagation compute similar solutions (i.e., sets of network connection strengths) to the coordinate transformation problem.

**Generalization**

A property that is often exhibited by artificial neural networks trained by examples is the ability to generalize from those examples, that is, to produce the correct output when presented with input patterns that were not in the training set. This property is of great theoretical and practical importance, as it dem- onstrates that the model network has not merely learned to associate patterns in the training set with their correct outputs on an independent basis, but has versions of our networks (Mixed and All All) and for networks with one and two hidden layers, as long as the output format is the same for the All-A and backpropagation networks being compared. The individual values of the weights are not the same after training by the three different procedures, but the overall structure of these weights is such that the two algorithms’ solutions to the coordinate transformation problem are functionally equivalent for the various network structures.

**Figure 10:** Response properties of hidden units in a backpropagation-trained network. Shown are transverse gain fields (a) and the receptive fields (b) encoded from the hidden units of a backpropagation network after it was trained to perform the coordinate transformation. Note the similarity between these response properties and those of the hidden units of All-A trained networks (Figs. 7, 9). Two of the receptive fields in (a, top right and middle right) are adapted from Zipser and Andersen (1988).
learned to perform the transformation implied in the training examples. In this case, mapping is the addition of two position vectors.

We tested two extensively trained networks for two types of problems: (1) All-Signals and (2) All-Numbers. One is the ability to perform the correct vector addition of new, random input patterns that code for the same output locations as the training set. As shown in Figure 12 (left), all three networks performed this task extremely well. The other generalization task required the trained networks to generalize their responses to new, random patterns for output locations that were not part of the training set. The performance of the networks on this task was considerably less than for the untrained nets, indicating that the networks generalized to a considerable extent.

Discussion

Choice of the Learning Algorithm

The choice of the algorithm used to update the network's connection strengths was the central issue of our experiments. The procedure used was designed to change the weights of a network in order to maximize some measure of performance in a supervised learning paradigm (for review, see Baro, 1987; Hinton, 1987; Lippmann, 1987; Anderson and Rosenfeld, 1988; McClelland and Rumelhart, 1988). An important class of learning algorithms concerns those that maximize the performance of the network in some objective function and adjust the weights to achieve this objective. This type of learning algorithm is generally called backpropagation learning.

Backpropagation is an important and powerful algorithm that can be adjusted to perform a wide variety of tasks. However, it is also important to note that the network's performance is limited by the complexity of the task at hand. In some cases, the network may not be able to learn a particular task, even if it is theoretically possible. This is because the network's learning algorithm is not always able to find a solution to the problem.

How the Network Learns

The learning process for backpropagation is gradient descent, the minimization of an error measure by following the negative of its gradient with respect to the weights. A priori there are no conceptual or experimental obstacles to envisioning the brain using this principle, too, given a plausible performance measure. The S-model algorithm as implemented in both of our network classes, also makes use of this general principle. While the backpropagation algorithm is the most obvious method for computing the exact error gradient for a given input pattern, the S-model algorithm computes only an estimate of that gradient. (Baro and Jordan, 1987). Units trained by the S-model, however, 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 nonbiological pathways. Due to the nature of their implementation, however, Aₜ units can be described as "jumping" their activity during learning so as to get an estimate of how the noise in activity affects the payoff they receive which then allows them to estimate the direction in weight space along which to change their weights in order to increase reinforcement.

This method allows Aₜ units to adjust their weights properly using only locally available information, it is more random in its search for a solution than backpropagation. These differences are obtained without loss of performance. In Figure 6, backpropagation's precise computation of the performance gradient tells the algorithm the exact manner in which to tune the network's weights. In the error is monotonically decreased, resulting in the smooth curve of Figure 6a. As this curve shows, the error falls quickly to a value close to 0. The problem with backpropagation (Fig.

Biological Plausibility of the Aₜ Algorithm

As we mentioned above, the S-model algorithm for our choice of a learning algorithm was a greater plausibility of biological implementation than the backpropagation algorithm. A possible weakness of the output set, however, that Aₜ units were not well described and did not map onto the biologival networks. Biological neural networks are known to be of the precise mechanisms of information processing in sensory systems, the most useful connoting a between artificial and biological neural networks is possibly to the description of abstract concepts in simplified models and the investigation of the possibility of implementing the S-model algorithm as an actual biological hardware. In other words, the Aₜ element was not designed by collecting scattered known facts of neurobiology and melding them into a computationally interesting unit capable of supervised learning, but rather as a simply, "neurally inspired" element with a few theoretically motivated features that make it interesting learning abilities. We will discuss biological plausibility, therefore, in its literal sense of suggesting that the abstract computing processes performed by the Aₜ units during learning become a natural model for biological neural mechanisms and partially demonstrated by experimental neuromathicists that the mechanisms differ from backpropagation networks.

The first and most salient element of Aₜ models, which aligns them with many neurobiological and psychological theories, is the ability to give the correct output coding for new output locations (Fig. 12, right), which is a more difficult task. Although all networks pro-

Cerebral Cortex July/Aug 1991, Vol. 1, No. 4, p. 302

A More Plausible Learning Network, Model of Area 7a+ in Monkeys et al.
learned to perform the transformation implied in the training examples. For instance, this mapping is the addition of two position vectors. We tested two extensively trained networks for two types of machine learning problems. One is the ability to perform the correct vector addition of new, random input patterns that code for the same output locations as the training set. As shown in Figure 12 (left), all three networks performed this task extremely well. The other generalization task required the trained networks to generate new output patterns (for input patterns not included in the training set). The ability to respond correctly to new patterns of this kind is a more difficult task. Although all networks produced a correct output, the error was considerably less than the error for the untrained nets, indicating that the networks generalized to a considerable extent.

Discussion

Choice of the Learning Algorithm

The choice of the algorithm used to update the net- work's connection strength was the central issue of our work. The external environment of procedures that change the weights of a network in order to maximize some measure of performance in a supervised learning paradigm for review, see Baro, 1985; Hinton, 1987; Lippman, 1987; Anderson and Rosenfeld, 1988; McClelland and Rumelhart, 1988. An important class of learning algorithms can be characterized by two attributes: the ability to change the weights of a network in order to maximize some measure of performance in a supervised learning paradigm. Figure 1 shows the network's connection strength as a function of performance in the environment. The ability of the network to learn from experience is defined by the rate of change of the weights with respect to the error signal. The rate of change of the connection strength is a function of the error signal and the rate of change of the output units, which is a function of the error signal and the rate of change of the weights. The error signal is the difference between the desired output and the actual output of the network. The rate of change of the weights is determined by the learning rule, which is a function of the error signal and the rate of change of the output units. The learning rule is determined by the network's architecture and the learning algorithm.

Biological Plausibility of the $A_{m}$ Algorithm

As we mentioned earlier, the most important goal of our choice of a learning algorithm was a greater plausi- bility of biological implementation than the back propagation algorithm. In order to assess this question, we analyzed the feedback signal, which is the only signal that enters the network at the output, so that $A_{m}$ networks were not able to simulate the most obvious feature of biological neural nets: the lateral connections. The most useful connection between artificial and biological neural networks is pres- ently limited to the description of abstract issues in simplified models and the investigation of the poss- ibility of simplifying the implementation of biological hardware. In other words, the $A_{m}$ element was not designed by collecting scattered known facts of neurobiology and molding them into a computationally interesting unit capable of supervised learning, but designed rather as a simple, "neurally inspired" element with a few theoretically motivated features that are important for interesting learning abilities. We will discuss the biological plausibility of the $A_{m}$ model with the help of specific features and its relationship to biological systems. The first and most salient element of the $A_{m}$ models, which aligns them with many neurobiological and psychological models, is the absence of a reinforcement signal. This has the attractive features of being computed from an average value of the error of the output units and of being transmitted in parallel of the error value to all the $A_{m}$ trained units in the network equally. This error could also be detected, for example, as a function of the change of the error signal with respect to the output of the net. This characteristic is the absence of an error signal with respect to the output of the network. This characteristic is the absence of an error signal with respect to the output of the network. This characteristic is the absence of an error signal with respect to the output of the network. This characteristic is the absence of an error signal with respect to the output of the network.
use such signals already exists. For example, the nu-
clear it is used in a broad system of cell-communication and error signals at all the connections in every layer above that connection (Rumelhart et al., 1986a).

The A₈ neuron, as we have mentioned, uses the A₉ rule only to train its hidden units. Its output units are trained by the delta rule (Eq. 4). Although this is the same learning rule as models of area 7a.

We have shown that the A₈ neuron can train a neural network to perform the same coordinate transformations as those described by Zipser and Anderson's model. We also found that the solution discovered by this algorithm is equivalent to that found by the in vivo coordinate transformations in the monkey (Zipser and Anderson, 1988). This solution gives hidden unit response properties (planar gain fields and low dimensional feature fields) very similar to those of area 7a neurons of code to spatial location. These response properties, therefore, are not a specific result of the backpropagation training procedure. The set of connections strengths computed by the A₈ neuron, moreover, is not a unique result because it is determined by an incorrect code. Other solutions, not involving planar gain fields or large receptive fields, can be constructed that would work for the training sets we used. It is striking, then, that A₈ neural and backpropagation produce this particular algorithm for computing coordinate transformations, and for the purpose of learning.

In a more detailed analysis of the model, we have shown that a second layer of hidden units can be added to the network without changing the response properties of the output units. Moreover, that these new networks are indeed capable of generalizing their coordinate transformation abilities to new input patterns. Both of these results strengthen the physiological significance of this model architecture. The former implies that relay elements—important and ubiquitous neural components of the nervous system—are not an obstacle to learning and allow similar solutions to develop. The latter establishes the important point that these model networks are indeed learning to perform the coordinate transformation task. They do not merely act as content-addressable memories, associating each input pattern in the training set with its correct output individually, but rather they are capable of abstracting from the training examples the transformation relationships between them, and applying it successfully to new pairs of retinal and eye position. This property has been observed before and is actually quite rare in the hidden layer compared to the input layer (S.-T. et al., 1987).

The neural network was trained using Hebbian learning (Hebb, 1949), that is, the proportionality of a change in synaptic strength to both presynaptic and postsynaptic firing rates. Learning requires one of the most attractive mechanisms for synaptic modification, both on theoretical (Lisker, 1989) and experimental grounds (Rushen et al., 1984; Keltson et al., 1986; Sejnowski et al., 1987; Brown et al., 1990). This is in contrast to backpropagation, in which changes in strength at a connection require information about the success or failure of activation at all the connections in every layer above that connection (Rumelhart et al., 1986a).

Simulation Results

The basic results of this study corroborate the validity, from a physiological perspective, of parallel networks which are trained to perform similar tasks as models of area 7a. We have shown that the A₈ neuron can train a neural network to perform the same coordinate transformation as models of area 7a neurons. Actually, any algorithm capable of performing gradient descent using a scalar reinforcement signal could be used to train a system in a similar context. This algorithm could be acceptable. Such algorithms are currently being developed (e.g., Gullapalli, 1989), and it would be a natural continuation of this work to try to apply them to networks modeling the neural system. The modeling methods in these algorithms involve the theoretical details of simultaneously updating the mean and variance of multivariate distribution of discrete continuous stochastic units. The present form of these algorithms is similar to that of the A₈ procedure for binary units. This is to consider that the extension of the concepts of supervised A₈ learning to networks with continuous output units will be a natural refinement that should not drastically change the types of solutions obtained.

Conclusions

We have shown that (1) the A₈ algorithm can train neural networks to compute coordinate transformations; (2) these networks are comparable to those obtained by backpropagation training; (3) in the process of learning this compu-
tation, the neurons develop gain fields and receptive fields qualitatively similar to those of area 7a neurons; (4) the solutions are equivalent to those computed by both the backpropagation and the A₈ algorithms. When applied to a class of parallel distributed net-
works, and that they can do so using mechanisms not excluded to the brain, perhaps suggested, by neurophysiologi-
cal evidence.

These results represent a step toward establishing the physiological validity of the architecture and gen-
eral learning principles of the model of area 7a intro-
duced by Zipser and Anderson. They show that phys-
ological principles can arise from a more plausible learning algorithm than backpropagation, thus sug-
gesting that the detailed processes by which neuronal ensembles learn must be viewed in their ultimate collective behavior. Abstract opti-
maximizing principles, such as gradient descent, may in-

---

Cerebral Cortex July/August 1991, V 4 N 4 365
use such signals already exists. For example, the nu-
ticulars to this specific system of chemical and electric
signals involving apparent to be related to behaviorally
related behavioral processes (Richardson et al., 1988). 
The reinforcement signal required by our model, of course,
would not have to be distributed on such a wide scale. 
This is easier to see if we think about a specific moth-
tor task, for example, the accuracy of a sacoda to a 
target, and thus be used only by one or a few ensembles 
cellular units in the system. Because a single signal, however,
would be valid for an entire group of neurons, there
would be no need to propagate it through special 
tracts to get it to specific units in the network of the entorhinal
This feature of the Α₄ neurons is more attractive 
than backpropagation's requirement for the ezero
grade propagation of error signals along specific path-
ways.

Besides not having to carry specific information to 
train individual neurons, the reinforcement signal used 
in our networks has the advantage that it can be inde-
pendent of any coordinate system. In backpropagation,
the information needed for the correct head-
centered locations as a vector in a cartesian 
reference frame. The Α₄ neurons, on the other hand, com-
piles its feedback signal from the average of the out-
put error's absolute value (Eq. 2, 5), which is a single 
number that can be derived from the comparison of 
reference and as well as coordinate systems.

Another "biological" feature of learning by Α₄ neurons is the use of information that is locally available to 
that unit in the input space, as in the synaptically driven, 
ferrokinetic form of Hebbian learning. The weight ad-
djusting equation for the 4th Α₄ neuron (Eq. 1) 
consists of two terms, one assigning the "reward" and 
the "penalty" portions, respectively, of the 
learning rule. These terms consist of three compo-
te 
ents: (1) the payoff signal, r, and the corresponding 
function of the mean probability of firing. The vari-
ability of spike trains recorded from cortical neurons also 
the intrinsic variability of the Α₄ neurons' responses to 
input signals is essential for the learning process.
that the network to filter out weights around the current 
set of values, thus sustaining the search for a beter 
solution. The noise provides the algorithm with the 
mechanism of obtaining information about local variations 
in reinforcement. By making successive incremental 
adjustments to the weights, the algorithm converges 
towards an estimate of the gradient of the 
reinforcement signal. In this manner the noise 
compensates, in a sense, for the scarcity of informa-
tion "in the scalar payoff signal. The re-
versing affects of the model, therefore, are not mere 
manifestations of robustness to noise. The Α₄ neurons 
demonstrate, rather, how a computational unit's out-
put variance can be used to achieve learning in a 
network that receives less than optimal feedback in 
information.

Simulation Results

The basic results of this study corroborate the validity, 
from a physiological perspective, of parallel networks 
with intrinsic, local information as models of area 7a.
We have shown that the Α₄ neurons can train a 
neural network to perform the same coordinate trans-
formations as those performed by the Α₄ neurons in 
the entorhinal model. We also found that the solution 
discovered by this algorithm is equivalent to that found 
by a network of neurons which, according to 
Anderson's analysis (Zipser and Anderson, 1988), this 
solution gives hidden unit response properties (pla-
nar gain fields and large visual receptive fields) very 
similar to those of area 7a neurons presumed to code 
for spatial location. These response properties, there-
fore, are not a specific result of the backpropagation 
training procedure. The set of connections strengths 
computed by the Α₄ neurons, moreover, is not a unique 
choice from among many that would do the job. 
Other solutions, not involving planar gain fields or 
large receptive fields, can be constructed that would 
work for the training sets we used. It is striking, then, 
that Α₄ neurons and backpropagation produce this particular 
algorithm for computing coordinate transformations, 
and not others.

In a more detailed analysis of the model, we have 
shown that a second layer of hidden units can be 
added to the network without changing the response 
properties of the output layer. Moreover, and that the model's 
comparability to those obtained by backpropagation 
training, (3) in the process of learning this 
computation, the Α₄ neurons develop gain fields and 
receptive fields qualitatively similar to those of area 7a 
neurons; (4) the solutions are equivalent to those 
comparable in these tasks. Furthermore, the computa-
tion models generalize appropriately. We have also pointed 
out a number of features of the Α₄ neuron model 
that bring it closer to the biological model, especially 
about biological learning. We must emphasize again 
that the focus of our interest at this point is not in 
how literally A₄ neuron models reproduce individual 
neurophysiological processes. It is rather the fact 
that these algorithms form a family of training 
procedures for cases like this; they are functional repre-
sentations of what happened when applied to a class of parallel distributed net-
works, and that they can do so using mechanisms and 
excluded networks, perhaps, is mic 
progress, however, to make the algorithm more re-
sistant to scale changes, or to program the network for 
more specifically reinforcement signal. Our use 
of a single scalar feedback signal could thus be 
viewed as a worst case scenario that does not exclude 
more specialized signals that may exist in 
biological systems.

Future Directions

It would be desirable to modify the Α₄ neurons so that 
it could learn in the presence of feedback from hidden 
units. Actually, any algorithm capable of 
performing gradient descent using a scalar reinforcement 
signal could in principle be translated to our model. It is 
acceptable. Such algorithms are currently being de-
eveloped (e.g., Gullapalli, 1989), and it would be a 
natural continuation of this work to try to apply them to 
networks modeling the visual system. The modeling 
capabilities in these algorithms involve the theoretical 
details of continuously updating the mean and variance 
of multiparameter discrete distributions, such as continuous 
stochastic processes. The present form of these algorithms 
is similar to that of the Α₄ procedure for binary units. 
It is conceivable that the extension of the concepts of 
backpropagation learning to networks with contin-
uous output units will be a natural refinement that 
should not drastically change the types of solutions 
obtained.

Conclusions

We have shown that (1) the Α₄ neuron can train 
neural networks to compute coordinate transforma-

tions; (2) the solutions obtained by them are 
comparable to those obtained by backpropagation 
training; (3) in the process of learning this 
computation, the Α₄ neurons develop gain fields and 
receptive fields qualitatively similar to those of area 7a 
neurons; (4) the solutions are equivalent to those 
comparable in these tasks. Furthermore, the computa-
tion models generalize appropriately. We have also pointed 
out a number of features of the Α₄ neuron model 
that bring it closer to the biological model, especially 
about biological learning. We must emphasize again 
that the focus of our interest at this point is not in 
how literally A₄ neuron models reproduce individual 
neurophysiological processes. It is rather the fact 
that these algorithms form a family of training 
procedures for cases like this; they are functional repre-
sentations of what happened when applied to a class of parallel distributed net-
works, and that they can do so using mechanisms and 
excluded networks, perhaps, is mic
stead be more important determinants of neuronal learning strategies, and it would be worthwhile to pursue such hypotheses with further theoretical and experimental studies.

Notes
The logistic function, which is a type of "squashing" function, has a sigmoidal shape and maps real-valued inputs into the interval 0 to 1, according to \( f(x) = 1 / (1 + e^{-x}) \). In our networks, this is the sum of the unit's inputs weighted by the corresponding connection strength, plus a bias.

2. The frequency of local minima was around 5% for backpropagation and approximately 1% for the \( A_N \) algorithms, in approximately 200 different simulations. One likely reason for the rather high frequency of local minima for backpropagation is the small number of hidden units in the network. The \( A_N \) networks were less affected by this parameter, mainly because the unit's output noise improved the network's chances of escaping local minima.

This work was supported by DHR Grant NB0145-89J1-1256 and NIH Grant EY05320 to R.A.A., by an acute grant the Siebens Foundation to M.O., and by NIH Medical Scientist Training Program Grant ST2C08775S to P.M. We thank Salamina G. Goodman for helpful discussion and for providing several computer programs.

Correspondence should be addressed to Dr. Richard A. Andersen, Department of Brain and Cognitive Sciences, E25-250, Massachusetts Institute of Technology, Cambridge, MA 02139.

References
stead be more important determinants of neuronal learning strategies, and it would be worthwhile to pursue such hypotheses with further theoretical and experimental studies.

Notes

1. The logistic function, which is a type of "squashing" function, has a sigmoidal shape and maps real-valued inputs into the interval 0 to 1, according to \( g(x) = \frac{1}{1 + e^{-ax}} \).

2. In our networks, this is the sum of the unit’s inputs weighted by the corresponding connection strength, plus a bias.

3. The frequency of local minima was around 5% for back-propagation and approximately 1% for the A_\text{ep} algorithm, in approximately 200 different simulations. One likely reason for the relatively high frequency of local minima for back-propagation is the small number of hidden units in the network. The A_\text{ep} networks were less affected by this parameter, mainly because the unit's output noise improved the network's chances of escaping local minima.

This work was supported by NIH Grant GM069114-02-JA1-JD6 and NIH Grant GM069114-02 to R.A. and, for a short time, the Sigma Corporation to M.J.L., and by NIH Medical Scientist Training Program Grant 5T32CA076775-10 to P.M. We thank Salas (I.), Goodman for helpful discussions and for providing several computer programs.

Correspondence should be addressed to Dr. Richard A. Andersen, Department of Brain and Cognitive Sciences, E25-350, Massachusetts Institute of Technology, Cambridge, MA 02139.

References


