NATO ASI Series

Advanced Science Institutes Series

A series presenting the results of activities sponsored by the NATO Science Committee, which aims at the dissemination of advanced scientific and echnological knowledge, with a view to strengthening links between scientific ommunities.

he Series is published by an international board of publishers in conjunction with e NATO Scientific Affairs Division

Life Sciences **Physics**	Plenum Publishing Corporation London and New York
Mathematical and Physical Sciences	D. Reidel Publishing Company Dordrecht, Boston, Lancaster and Tokyo
Behavioural and Social Sciences **Applied Sciences**	Martinus Nijhoff Publishers Boston, The Hague, Dordrecht and Lancaster
Computer and Systems Sciences **Ecological Sciences** **Cell Biology**	Springer-Verlag Berlin Heidelberg New York London Paris Tokyo

ries F: Computer and Systems Sciences Vol. 41

Neural Computers

Edited by

Rolf Eckmiller

Department of Biophysics
University of Düsseldorf
Universitätsstraße 1
D-4000 Düsseldorf, FRG

Christoph v. d. Malsburg

Department of Neurobiology
Max-Planck-Institute of
Biophysical Chemistry
Nikolausberg
D-3400 Göttingen, FRG

Springer-Verlag
Berlin Heidelberg New York London Paris Tokyo
Published in cooperation with NATO Scientific Affairs Division

NEURAL NETWORK LEARNING ALGORITHMS

Terrence J. Sejnowski
Department of Biophysics
Johns Hopkins University
Baltimore, MD 21218

ABSTRACT

The earliest network models of associative memory were based on correlations betwee input and output patterns of activity in linear processing units. These models have sever features that make them attractive: The synaptic strengths are computed from informatio available locally at each synapse in a single trial; the information is distributed in a larg number of connection strengths, the recall of stored information is associative, and the network can generalize to new input patterns that are similar to stored patterns. There are als severe limitations with this class of linear associative matrix models, including interferenc between stored items, especially between ones that are related, and inability to make decisions that are contingent on several inputs. New neural network models and neural networ learning algorithms have been introduced recently that overcome some of the shortcoming of the associative matrix models of memory. These learning algorithms require many training examples to create the internal representations needed to perform a difficult task and generalize properly. They share some properties with human skill acquisition.

1. INTRODUCTION

Processing in neurons can be very complex, though within the basic limitations o speed and accuracy imposed by the biophysical properties of ions and membranes. Integration of information in dendrites is often nonlinear. There are many types of neurons that have highly specific patterns of connectivity; some are primarily inhibitory, others are primarily excitatory, and synaptic strengths are variable on many time scales. Finally, the nervous system has many different nuclei and many cortical areas that have different structural motifs as well as different functions. How much of the details of neural processing must be included in a model? This depends on the level under investigation (Churchland et al., 1988). Biophysical properties may be crucial when modeling synaptic plasticity, but only a general rule for modification may be needed to model information storage at the circuit level. The style of processing and memory, such as the degree to which information is localized or distributed in the network, could well be general properties while the actual codes used are probably specific to the detailed circuits.

As a first step toward understanding real neural networks, we study network models constructed from simple processing units that have only the most basic properties of neurons and attempt to explore their computational capabilities: What are the possible ways to represent sensory information in a collection of these units? What are the computational capabilities of different patterns of connectivity in the network? What computations can the network not perform? Even the simplest networks have complex behaviors that are not easy

NATO ASI Series, Vol. F41
Neural Computers
Edited by R. Eckmiller and Ch. v. d. Malsburg

to describe analytically, so much of the research is empirical and exploratory. Also, there are so many architectures — the number of layers, feedback between layers, and local patterns of connectivity — that much guidance is needed from the general organization of cortical circuits, such as the columnar organization of cerebral cortex and the hierarchical arrangements of cortical mappings (Sejnowski, 1986).

2. ASSOCIATIVE MATRIX MODELS

The goal of linear associative matrix models (Steinbuch, 1961; Anderson, 1970; Kohonen, 1972) was to perform content-addressable recall of information stored as vectors. Given an input vector ι_b and an associated output vector o_a, the correlation matrix is defined as:

$$w_{ij} = \varepsilon\, o_i \iota_j \tag{1}$$

where ε is the strength of the association and w_{ij} represents a linear transformation between input vectors and output vectors. If ι_j is identified with the rate of firing of the jth presynaptic element and o_i is identified with rate of firing of the ith postsynaptic element, then K_{ij} can be computed after modifying the synapses between the input and output neurons according to the learning rule suggested by Hebb (1949), which states that a plastic synapse should increase in strength whenever there is a simultaneous presynaptic spike and a postsynaptic spike. An important property of the correlation matrix is that it depends only on information that is available locally at a synapse. Nonlocal modification rules that require information from disparate parts of a network are more difficult to implement.

Hebbian synaptic plasticity is probably the simplest local rule that can be used for associative storage and recall of information. Evidence supporting Hebbian plasticity has recently been found in the hippocampus (Kelso, et al., 1986) and detailed correlation matrix models of the hippocampus are now being explored (Lynch, 1986; Rolls, 1986). However, there are many other uses for Hebbian synaptic plasticity, such as plasticity during development (Linsker, 1986), unsupervised learning (Sutton & Barto, 1981; Tesauro, 1986; Finkel & Edelman, 1985), and very rapid changes in the topology of a network (von der Malsburg & Bienenstock, 1986). As a consequence, experimental evidence for Hebbian modification of synaptic strength does not necessarily imply associative storage.

Numerous variations have been proposed on the conditions for Hebbian plasticity (Levy, et al., 1984). One problem with any synaptic modification rule that can only increase the strength of a synapse is the eventual saturation of the synaptic strength at its maximum value. Nonspecific decay is one solution to this problem. Sejnowski (1977a, 1977b) has suggested that specific decreases in the strength of a plastic synapse should be considered, and proposed that the change in strength of a plastic synapse should be proportional to the covariance between the presynaptic firing and postsynaptic firing:

$$w_{ij} = \varepsilon\, (o_i - \bar{o}_i)\,(\iota_j - \bar{\iota}_j) \tag{2}$$

where $\bar{o}_i$ is the average firing rate of the output neuron and $\bar{\iota}_j$ is the average firing rate of the input neuron. (See also Chauvet, 1986) According to this modification rule, the strength of the synapse should increase if the firing of the presynaptic and postsynaptic elements are positively correlated, decrease if they are negatively correlated, and remain unchanged if they are uncorrelated. Evidence for a decrease in the strength of synapses in the hippocampus under the predicted conditions has recently been reported by Levy, et al. (1983). Similar modification rules have also been suggested for plasticity during development (Cooper et al., 1979; Bienenstock et al., 1982).

Improvements have recently been made to associative matrix models by introduc feedback connections, so that they are autoassociative, and by making them nonlir (Anderson & Mozer, 1981; Sejnowski, 1981; Hopfield, 1982; Kohonen, 1984; Golden, 19 Toulouse et al., 1986). However, this class of models still has a severe computational lim tion in that all the processing units in the network are constrained by either the inputs or outputs, so that there are no free units that could be used to form new internal represer tions. What representations should be used for the input units and output units if the netw is deeply buried in association cortex? Some other principles must be specified for form internal representations. Nevertheless, given that good representations already exist, associative matrix model is still a viable one for the fast storage of novel events and ite (Hinton & Anderson, 1981).

3. NONLINEAR NETWORKS

The output of the model neuron introduced by McCulloch and Pitts (1943) could o take on the values of 0 or 1, like the all-or-none action potential. This binary model does take into account the graded responses of neurons, which can be expressed as an average r of firing. There are two ways make the output of the processing unit graded. First, the o put of the processing unit can be made probabilistic, with a probability proportional to average rate of firing. Secondly, the output of a processing unit can be made a real numb between 0 and 1. Both of these possibilities will be discussed in this chapter.

The firing rate of neuron as a function of injected current has a threshold and saturat at some maximum firing rate. A simple model for this function is the sigmoid:

$$s_i = P(E_i) = \frac{1}{1 + e^{-\sum\limits_j w_{ij} s_j}}, \qquad (3$$

where s_i is the output of the ith unit, and w_{ij} is the weight from the jth to the ith unit. Th weights can have positive or negative real values, representing an excitatory or inhibitor influence. In addition to the weights connecting them, each unit also has a threshold and i some learning algorithms the thresholds can vary.

In a network of processing units, a subset receives information from outside the ne work while another subset provides the output from the network. The networks that we ca study are small and should be considered small circuits embedded in a larger system. Pat terns of activity in the group of input units are transformed into patterns of activity in th output units by direct connections and through connections with additional internal units tha play the role of interneurons. In general, it is very difficult to analyze the performance an computational capabilities of nonlinear network models. By making restrictions on the con nectivity it is possible to make progress.

In a network with feedback the output units may reverberate without settling down to a stable output. In some cases oscillations may be desirable, but otherwise special provisions must be made to suppress the them. One method that has been thoroughly explored is the use of symmetric connectivity. Networks with reciprocal symmetric connections, first introduced by Hopfield (1982) in the context of binary-valued processing units, were the starting point for the study of learning algorithms in Boltzmann machines by Hinton and Sejnowski (1983). Another method, extensively studied by Grossberg (1983), is the use of lateral shunting inhibition. But it is easiest to avoid oscillations by not considering any feedback connections.

In a feedforward network there is no dynamic feedback so that information can only flow from the input layer to the output layer. The simplest class of feedforward networks are ones that have no internal or "hidden" units. In this case each output unit acts independently in response to input patterns in its "receptive field", defined here as the group of input units which drive the output unit, in analogy with the concept of a receptive field for sensory neurons. The output unit is most strongly driven by patterns of activity in its receptive field that are congruent with the excitatory connections and that avoid the inhibitory ones.

A very simple learning procedure exists for automatically determining the weights in a single-layer feedforward network. It is an incremental learning procedure that requires a teacher to provide the network with examples of typical input patterns and the correct outputs; with each example the weights in the network are slightly altered to improve the performance of the network. If a set of weights exists that can solve the classification problem, then convergence theorems guarantee that such a set of weights will be found. These learning procedures are error-correcting in the sense that only information about the discrepancy between the desired outputs provided by the teacher and the actual output given by the network is used to update the weights. The LMS algorithm of Widrow and Hoff (1960) applies to units that have continuous-valued outputs, and the perceptron learning algorithm of Rosenblatt (1959) applies to binary-valued units. These error correction algorithms require that the weight from input unit s_j to the *i* th output unit should be altered by

$$\Delta w_{ij} = \varepsilon \, (s_i^* - s_i) \, s_j . \qquad (4)$$

where s_i^* is the desired output, s_i is the actual output, and ε is the rate of learning. On each learning step the squared error averaged over all input patterns is reduced.

There is an interesting relationship between this error-correcting procedure and the Rescorla-Wagner theory for classical conditioning. Rescorla and Wagner (1972) state that: "Organisms only learn when events violate their expectations. Certain expectations are built up about the events following a stimulus complex; expectations initiated by the complex and its component stimuli are then only modified when consequent events disagree with the composite expectation." Thus it is the difference between the expected and actual outcomes that determine whether strengths are modified. Sutton and Barto (1981) have shown that the mathematical formalism introduced by Rescorla and Wagner is identical to the Widrow-Hoff LMS algorithm.

Recently, Gluck and Bower (1987) have applied the LMS algorithm to category learning in humans. In three experiments, subjects learned to categorize diseases in hypothetical patients from patterns of symptoms. The adaptive network model was a better predictor of human performance than probability matching, exemplar retrieval, or simple prototype matching. The model correctly predicted a counterintuitive phenomenon called "base-rate neglect" that has been frequently observed in studies of liklihood judgments: When one disease is far more likely than another, the model predicts that subjects will overestimate the diagnostic value of the more valid symptom for the rare disease. Thus, the subjects consistently overestimated the degree to which evidence that was representative or typical of a rare event was actually predictive of it (Kahneman & Tversky, 1972).

The patterns that can be correctly classified with a one-layer network are limited to those that are geometrically equivalent to regions of a vector space bounded by a plane (Minsky & Papert, 1969). Single-layer networks are severely limited in the difficulty of the problem that they can solve, but this deficiency can be partially overcome by preprocessing the inputs through a layer of units that serve as feature detectors so that the information needed to solve the problem is made explicitly available (Rosenblatt, 1959; Gamba, et al., 1961). The required features may be different for each problem.

One problem with single-layer networks is the lack of internal degrees of freedom. Can the learning algorithm be generalized to networks with more than one layer of weights? If so, then the need to hand-code the features for each problem would be alleviated, and much more difficult problems could be solved by the same type supervised learning paradigm. It had been thought for many years that such a learning algorithm was not possible for multilayered networks (Minsky and Papert, 1969, p. 232; see also Arbib, 1987).

4. NETWORK MODELS WITH HIDDEN UNITS

Adding a single intermediate layer of hidden units between the input and output layers suffices to perform any desired transformation (Palm, 1979). However, the number of hidden units required may be very large. In practice, only a small subset of all possible transformations are ever needed and only a small number of hidden units are available. The challenge is to find the appropriate set of hidden units for each problem. One possibility is to have the network discover the proper features without supervision from a teacher. There are several unsupervised learning procedures that can automatically model structure from the environment (Kohonen, 1984; Grossberg, 1976; Rumelhart & Zipser, 1985; Pearlmutter & Hinton, 1986). One problem with unsupervised learning is that all the hidden units may discover the same features. Competition through mutual inhibition is one solution that enforces diversity (Feldman, 1982). Another problem is that not all the structure in the inputs may be relevant to the solution of a particular problem. Feedback of information from the environment about the desired performance is needed.

One class of supervised learning algorithms for multilayered networks uses reinforcement signals from a teacher that tell the network whether or not the output is correct (Sutton & Barto, 1981; Barto, 1986; Klopf, 1986; Tesauro, 1986; Gluck & Thompson, 1986). This is the minimum amount of information needed to help direct the hidden units toward good features, but there is so little information that the networks improve slowly and hesitatingly. Recently a new class of algorithms have been discovered that directly generalize the class of error-correcting learning procedures to multilayered networks. Two examples will be reviewed here: The Boltzmann machine and back-propagation. [See also Arbib (1987) for a review that includes a valuable historical perspective on earlier work.]

Boltzmann machines. Hinton & Sejnowski (1983, 1986) introduced a stochastic network architecture, called the Boltzmann machine, for solving optimization problems (Marr & Poggio, 1976; Ballard et al., 1983; Hopfield & Tank, 1986). The processing units in a Boltzmann machine are binary and are updated probabilistically using the output function in Eq. 1 to compute the probabilities. As a consequence, the internal state of a Boltzmann machine fluctuates even for a constant input pattern. The amount of fluctuation is controlled by a parameter that is analogous to the temperature of a thermodynamic system. Fluctuations allow the system to escape from local traps into which it would get stuck if there were no noise in the system. All the units in a Boltzmann machine are symmetrically connected: this allows an "energy" to be defined for the network and insures that the network will relax to an equilibrium state which minimizes the energy (Hopfield, 1982). Smolensky (1983) has studied the same architecture using "harmony" as the global function, which is the negative of energy.

The Boltzmann machine has been applied to a number of constraint satisfaction problems in vision, such as figure-ground separation in image analysis (Sejnowski & Hinton, 1987; Kienker et al., 1986), and generalizations have been applied to image restoration (Geman & Geman, 1984) and binocular depth perception (Divko & Schulten, 1986). The number of times that the network must be updated to reach an optimal solution can be very large when the units are stochastic; architectures with continuous-valued units can converge to near optimal solutions much more quickly (Hopfield, 1984; Hopfield & Tank, 1985, Hopfield & Tank, 1986).

Boltzmann machines have an interesting learning algorithm that allows "energy landscapes" to be created through training by example. Learning in a Boltzmann machine has two phases. In the training phase a binary input pattern is imposed on the input group as well as the correct binary output pattern. The system is allowed to relax to equilibrium at a fixed "temperature" while the inputs and outputs are held fixed. In equilibrium, the average fraction of the time a pair of units is on together, the co-occurrence probability p_{ij}^+, is computed for each connection. In the test phase the same procedure is followed with only the input units clamped, and the average co-occurrence probabilities, p_{ij}^-, are again computed. The weights are then updated according to:

$$\Delta w_{ij} = \varepsilon \, (p_{ij}^+ - p_{ij}^-), \qquad (5)$$

where ε controls the rate of learning. A co-occurrence probability is related to the correlation between the firing or activation of the presynaptic and postsynaptic units and can be implemented by a Hebb synapse. In the second phase, however, the change in the synaptic strengths is anti-Hebbian since it must decrease with increasing correlation. Notice that this procedure is also error- correcting, since no change will be made to the weight if the two probabilities are the same. The perceptron learning procedure follows as a special case of the Boltzmann learning algorithm when there are no hidden units and the probability function reduces to a step function.

The Boltzmann learning algorithm has been applied to a variety of problems, such as the bandwidth compression (Ackley, Hinton & Sejnowski, 1985), learning symmetry groups (Sejnowski et al., 1986), and speech recognition (Prager, Harrison & Fallside, 1986). One of the practical limitations of simulating a Boltzmann machine on a conventional digital computer is the excessive time required to come to equilibrium and collect statistics. A special-purpose VLSI chips is being designed to speed up the learning (Alspector & Allen, 1986). Recently, a mean-field theory for the Boltzmann machine has been introduced for which learning is an order of magnitude faster (Anderson, 1987).

Back-propagation. Another error-correcting learning procedure, called error back-propagation, generalizes the Widrow-Hoff algorithm to multilayered feedforward architectures (Rumelhart, et al., 1986; Parker, 1986; Le Cun, 1985). The back-propagation learning algorithm has been applied to many problems, including knowledge representation in semantic networks (Hinton, 1986), text-to-speech (Sejnowski & Rosenberg, 1987), sonar target identification (Gorman & Sejnowski, 1987), backgammon (Tesauro & Sejnowski, 1986), and predicting the secondary structure of globular proteins (Qian & Sejnowski, 1988).

Neither the Boltzmann machine nor the error back-propagation scheme are meant as literal models of real neural circuitry. They are also quite different from each other — The Boltzmann machine uses binary stochastic units in a symmetric network while back-propagation uses real-valued deterministic units in a feedforward network — but both architectures have learning algorithms that depend on gradient descent in the space of weights, which can have a very high dimensionality. The class of gradient descent algorithms for learning in large networks may have general properties that are already present in the simplest members. Other more elaborate gradient descent learning algorithms, which are more biologically plausible, are also being explored.

5. CONCLUSIONS

Error correction learning procedures such as the Boltzmann machine learning algorithm and error back-propagation require many repetitions of the training examples. It also takes long practice to become an expert in domains such as playing chess, proving mathematical theorems, and 17th century intellectual history. Slow learning allows efficient internal representations to be built up amongst the hidden units. Once these representations have been formed, they can be used to perform fast associative storage and recall of facts that are domain specific and use the structure of the domain for analogical reasoning about other domains. Learning systems will need a variety of adaptive mechanisms for storing, reorganizing, and retrieving experiences.

Acknowledgments: Preparation of this chapter was supported by grants from the National Science Foundation, Seaver Institute, and the Air Force Office of Scientific Research.

REFERENCES

Ackley, D. H., Hinton, G. E. & Sejnowski, T. J., 1985. A learning algorithm for Boltzmann machines, Cognitive Science 9, 147-169.

Alspector, J. & Allen, R. B., A VLSI model of neural nets, Bellcore Technical Memorandum TM ARH002688.

Anderson, J. R., 1987, Development of an analog neural network model of computation, University of Texas Department of Computer Science Technical Report TR-87-15.

Arbib, M. A., 1987, Brains, Machines & Mathematics, 2nd edition, New York: McGraw-Hill Press

Anderson, J. A., 1970, Two models for memory organization using interacting traces, Mathematical Biosciences 8, 137-160.

Anderson, J. A. & Mozer, M. C., 1981, Categorization and selective neurons, In: Parallel models of associative memory, Hinton, G. E. & Anderson, J. A., (Eds.) Hillsdale, N. J.: Erlbaum Associates.

Ballard, D. H., Hinton, G. E., & Sejnowski, T. J., 1983. Parallel visual computation, Nature 306: 21-26.

Barto, A. G., 1985, Learning by statistical cooperation of self-interested neuron-like computing elements, Human Neurobiology 4, 229-256.

Bienenstock, E. L., Cooper, L. N. & Munro, P. W., 1982, Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex, Journal of Neuroscience 2, 32-48.

Chauvet, G., 1986, Habituation rules for a theory of the cerebellar cortex, Biol. Cybernetics 55, 201-209.

Churchland, P. S., Koch, C. & Sejnowski, T. J., 1988, What is computational neuroscience?, In: Computational Neuroscience, E. Schwartz (Ed.), Cambridge: MIT Press.

Cohen, M. A. & Grossberg, S., 1983, Absolute stability of global pattern formation and parallel memory storage by competitive neural networks, IEEE Transaction on Systems, Man and Cybernetics, 13, 815-825.

Cooper, L. N., Liberman, F. & Oja, E., 1979, A theory for the acquisition and loss of neuron specificity in visual cortex, Biological Cybernetics 33, 9-28.

Divko, R. & Schulten, K., 1986, Stochastic spin models for pattern recognition, In: Neural Networks for Computing, AIP conference Proceedings 151, J. S. Denker, Ed., New York: American Institute of Physics, 129-134.

Feldman, J. A., 1982, Dynamic connections in neural networks, Biological

Cybernetics 46, 27-39.

Feldman, J. A., 1986, Neural representation of conceptual knowledge, Technical Report TR-189, University of Rochester Department of Computer Science.

Feldman, J. A. & Ballard, D. H., 1982. Connectionist models and their properties, Cog. Sci. 6: 205-254.

Finkel, L. H. & Edelman, G. M., 1985, Interaction of synaptic modification rules within population of neurons, Proceedings of the National Academy of Sciences USA, 82, 1291-1295.

Gamba, A. L., Gamberini, G., Palmieri, G., & Sanna, R., 1961, Further experiments with PAPA, Nuovo Cimento Suppl., No. 2, 20, 221-231.

Geman, S. & Geman, D., 1984, Stochastic relaxation, Gibbs distributions, and the Baysian restoration of images, IEEE Transactions on Pattern Analysis and Machine Intelligence 3, 79-92.

Gluck, M. A. & Bower, G. H., 1987, From conditioning to category learning: An adaptive network model, in preparation.

Gluck, M. A. & Thompson, R. F., 1987, Modeling the neural substrates of associative learning and memory: A computational approach, Psychological Review

Golden, R. M., 1987, The "brain-state-in-a-box" neural model is a gradient descent algorithm, Journal of Mathematical Psychology, in press.

Gorman, R. P. & Sejnowski, T. J., 1988, Learned classification of sonar targets using a massively-parallel network, IEEE Trans. Acous. Speech Signal Proc. (submitted).

Grossberg, S., 1976, Adaptive pattern classification and universal recoding: I: Parallel development and coding of neural feature detectors. Biological Cybernetics 23, 121-134.

Hebb, D. O., 1949, Organization of Behavior, New York: John Wiley & Sons.

Hinton, G. E., 1986, Learning distributed representations of concepts, Proceedings of the Eighth Annual Conference of the Cognitive Science Society, Hillsdale, New Jersey: Erlbaum, 1-12.

Hinton, G. E. & Anderson, J. A., 1981, Parallel models of associative memory, Hillsdale, N. J.: Erlbaum Associates.

Hinton, G. E. & Sejnowski, T. J., 1983. Optimal perceptual inference, Proceedings of the IEEE Computer Society Conference on Computer Vision & Pattern Recognition, Washington, D. C., 448-453.

Hinton, G. E. & Sejnowski, T. J., 1986, Learning and relearning in Boltzmann machines, In: McClelland, J. L. & Rumelhart, D. E., 1986, Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 2: Psychological and Biological Models. Cambridge: MIT Press, 282-317.

Hopfield, J. J., 1982, Neural networks and physical systems with emergent collective computational abilities, Proceedings of the National Academy of Sciences USA 79: 2554-2558.

Hopfield, J. J., 1984, Neurons with graded response have collective computation abilities, Proceedings of the National Academy of Sciences USA 81: 3088-3092.

Hopfield, J. J. & Tank, D., 1985. "Neural" computation of decision in optimization problems, Biol. Cybernetics, 52, 141-152.

Hopfield, J. J. & Tank, D., 1986, Computing with neural circuits: A model, Science 233, 624-633.

Kahneman, D. & Tversky, A., 1972, Subjective probability: A judgement of representativeness, Cognitive Psychology 3, 430-454.

Kelso, S. R., Ganong, A. H., & Brown, T. H., 1986, Hebbian synapses in hippocampus, Proceedings of the National Academy of Sciences USA, 83 5326-5330.

Kienker, P. K., Sejnowski, T. J., Hinton, G. E. & Schumacher, L. E., 1986, Separati figure from ground with a parallel network, Perception 15, 197-216.

Klopf, A. H., 1986, A drive-reinforcement model of single neuron function: An alte native to the Hebbian neuronal model, In: Neural Networks for Computing, J. S. Denk (Ed.), New York: American Institute of Physics, 265-270.

Kohonen, T., 1970, Correlation matrix memories, IEEE Transactions on Computer C-21, 353-359.

Kohonen, T., 1984, Self-Organization and Associative Memory, New York: Springe Verlag

Le Cun, Y., 1985, A learning procedure for asymmetric network, Proceedings of Cog nitiva 85, 599-604. Paris.

Levy, W. B., Anderson, J. A. & Lehmkuhle, W., 1984, Synaptic Change in the Ne vous System, Hillsdale, New Jersey: Erlbaum.

Levy, W. B., Brassel, S. E., & Moore, S. D., 1983, Partial quantification of the associa tive synaptic learning rule of the dentate gyrus, Neuroscience 8, 799-808.

Linsker, R., 1986, From basic network principles to neural architecture: Emergence o orientation columns, Proceedings of the National Academy of Sciences USA, 83, 8779-8783

Lynch, G., 1986, Synapses, Circuits, and the Beginnings of Memory, Cambridge: MI Press.

McClelland, J. L. & Rumelhart, D. E., 1986, Parallel Distributed Processing: Explora tions in the Microstructure of Cognition. Vol. 2: Psychological and Biological Models Cambridge: MIT Press.

Marr, D. & Poggio, T, 1976, Cooperative computation of stereo disparity, Science 194 283-287.

McCulloch, W. S. & Pitts, W. H., 1943, A logical calculus of ideas immanent in ner vous activity, Bull. Math. Biophysics, 5, 115-133.

Minsky, M. & Papert, S., 1969. Perceptrons, Cambridge: MIT Press.

Palm, G., 1979, On representation and approximation of nonlinear systems, Part II Discrete time, Biological Cybernetics, 34, 49-52.

Parker, D. B, 1986, A comparison of algorithms for neuron-like cells, In: Neural Networks for Computing, J. S. Denker (Ed.), New York: American Institute of Physics, 327-332.

Pearlmutter, B. A. & Hinton, G. E., 1986, G-Maximization: An unsupervised learning procedure for discovering regularities, In: Neural Networks for Computing, J. S. Denke (Ed.), New York: American Institute of Physics, 333-338.

Prager, R. W., Harrison, T. D, & Fallside, F., 1986, Boltzmann machines for speech recognition, Computer Speech and Language 1, 3-27 (1987)

Qian, N. & Sejnowski, T. J., 1988, Predicting the secondary structure of globular proteins using neural network models, J. Molec. Biol. (submitted).

Rescorla, R. A. & Wagner, A. R., 1972, A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and non- reinforcement. In: A. H. Black & W. F. Prokasy (Eds.), Classical Conditioning II: Current Research and Theory, New York: Appleton-Crofts.

Rolls, E. T., 1986, Information representation, processing and storage in the brain: Analysis at the single neuron level, In: Neural and Molecular Mechanisms of Learning, Berlin: Springer-Verlag.

Rosenberg, C. R. & Sejnowski, T. J., 1986, The spacing effect on NETtalk, a massively-parallel network, Proceedings of the Eighth Annual Conference of the Cognitive Science Society, Hillsdale, New Jersey: Lawrence Erlbaum Associates 72-89.

Rosenblatt, F., 1959, Principles of Neurodynamics, New York: Spartan Books.

Rumelhart, D. E., Hinton, G. E. & Williams, R, J., 1986. Learning internal representations by error propagation, In: Rumelhart, D. E. & McClelland, J. L., Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1: Foundations. Cambridge: MIT Press.

Rumelhart, D. E. & McClelland, J. L., 1986, Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1: Foundations. Cambridge: MIT Press.

Rumelhart, D. E. & Zipser, D., 1985, Feature discovery by competitive learning, Cognitive Science 9, 75-112.

Sejnowski, T. J., 1977a, Statistical constraints on synaptic plasticity, J. Math. Biology 69, 385-389.

Sejnowski, T. J., 1977b, Storing covariance with nonlinearly interacting neurons, J. Math. Biology 4, 303-321.

Sejnowski, T. J., 1981, Skeleton filters in the brain, In: Parallel models of associative memory, Hinton, G. E. & Anderson, J. A., (Eds.) Hillsdale, N. J.: Erlbaum Associates.

Sejnowski, T. J., 1986, Open questions about computation in cerebral cortex, In: McClelland, J. L. & Rumelhart, D. E., Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 2: Psychological and Biological Models. Cambridge: MIT Press, 372-389.

Sejnowski, T. J. & Hinton, G. E., 1987, Separating figure from ground with a Boltzmann Machine, In: Vision, Brain & Cooperative Computation, Eds. M. A. Arbib & A. R. Hanson (MIT Press, Cambridge).

Sejnowski, T. J., Kienker, P. K. & Hinton, G. E., 1986, Learning symmetry groups with hidden units: Beyond the perceptron, Physica 22D, 260-275.

Sejnowski, T. J. & Rosenberg, C. R., 1986, NETtalk: A parallel network that learns to read aloud, Johns Hopkins University Department of Electrical Engineering and Computer Science Technical Report 86/01

Smolensky, P., 1983, Schema selection and stochastic inference in modular environments, In: Proceedings of the National Conference on Artificial Intelligence, Los Altos, California: William Kauffman.

Steinbuch, K., 1961, Die lernmatrix, Kybernetik 1, 36-45.

Sutton, R. S. & Barto, A. G., 1981, Toward a modern theory of adaptive networks: Expectation and prediction, Psychological Review 88, 135-170.

Tesauro, G., 1986, Simple neural models of classical conditioning, Biological Cybernetics 55, 187-200.

Tesauro, G. & Sejnowski, T. J., 1987, A parallel network that learns to play backgammon, Artificial Intelligence (submitted).

Toulouse, G., Dehaene, S., & Changeux, J.-P., 1986, Spin glass model of learning by selection, Proceedings of the National Academy of Sciences USA, 83, 1695-1698.

von der Malsburg, C., & Bienenstock, E., 1986, A neural network for the retrieval of superimposed connection patterns, In: Disordered Systems and Biological Organization, F. Fogelman, F. Weisbuch, & E. Bienenstock, Eds., Springer-Verlag: Berlin.

Widrow, G. & Hoff, M. E., 1960, Adaptive switching circuits, Institute of Radio Engineers Western Electronic Show and Convention, Convention Record 4, 96-194.

Wilshaw, D., 1981, Holography, associative memory, and inductive generalization, In: Hinton, G. E. & Anderson, J. A., Parallel Models of Associative Memory, Hillsdale, New Jersey: Lawrence Erlbaum Associates.