Counterpropagation networks

Robert Hecht-Nielsen

By combining Kohonen learning and Grossberg learning a new type of mapping neural network is obtained. This counterpropagation network (CPN) functions as a statistically optimal self-programming lookup table. The paper begins with some introductory comments, followed by the definition of the CPN. Then a closed-form formula for the error of the network is developed. The paper concludes with a discussion of CPN variants and comments about CPN convergence and performance. References and a neurocomputing bibliography with a combined total of eighty entries are provided.

I. Introduction

Of all the practical information processing operations that neural networks can currently carry out, one of the most useful is the ability to learn a mathematical mapping by self-organization in response to examples of the mapping's action. Typically, one generates a set of examples $(\mathbf{x}_1,\mathbf{y}_1)$, $(\mathbf{x}_2,\mathbf{y}_2)$, ... of the action of a function ϕ, where $\mathbf{y}_i = \phi(\mathbf{x}_i)$ or $\mathbf{y}_i = \phi(\mathbf{x}_i) + \mathbf{n}$ (where $\mathbf{n}$ is a stationary noise process). These examples statistically define the desired input/output relationship.

A network that self-organizes itself to implement an approximation to a function or mapping is called a mapping neural network. Currently, the most popular mapping neural network is the backpropagation network of Rumelhart.[1] The backpropagation network has been shown to be capable of implementing approximations to a variety of mappings from $\mathbf{R}^n$ to $\mathbf{R}^m$. Although most applications of backpropagation to date have been to mappings that have binary input and output vectors (i.e., each vector component is essentially either one or zero), there is ample evidence that backpropagation also works for at least some nonbinary vector mappings.

In this paper a new type of mapping neural network, called the counterpropagation network (CPN), is introduced. It is shown that (under nonpathological conditions) this network will self-organize a near-optimal (in the sense that the entries in the table are statistically equiprobable) lookup table approximation to the mapping used to generate its data. The method works equally well for both binary and continuous vector mappings. It is shown that for a sufficiently large network the mapping approximation can be made essentially as accurate as desired. The counterpropagation network architecture is a combination of a portion of the self-organizing map of Kohonen[2] and the outstar structure of Grossberg.[3]

Although most pure mapping network problems are best attacked by use of backpropagation, those applications that require statistically equiprobable feature vectors or where a lookup table structure is desirable should be approached using the counterpropagation network. Problems such as vector quantization code development, high-dimensional probability density function estimation and characterization, and pattern recognition for continuously variable pattern vectors are often best solved using counterpropagation networks.

II. Counterpropagation Network

Figure 1 presents the topology of the CPN and Fig. 2 presents the transfer functions and learning laws of each of the five layers of the network. The basic idea is that, during adaptation, pairs of example vectors $(\mathbf{x},\mathbf{y})$ (both assumed to be of unit length) are presented to the network at layers 1 and 5, respectively. These vectors then propagate through the network in a counterflow manner to yield output vectors $\mathbf{x}'$ and $\mathbf{y}'$ that are intended to be approximations of $\mathbf{x}$ and $\mathbf{y}$. Thus the name counterpropagation.

As shown in Fig. 1, the full $\mathbf{x}$ and $\mathbf{y}$ input vectors are supplied to each processing element of layer 3, whereas the processing elements of layers 2 and 4 only get their corresponding component of these vectors. The processing elements of layers 2 and 4 also receive inputs from each processing element of layer 3 (the layer 3 processing element output signals, of which there are N, are labeled z_i).

When an $\mathbf{x}$ and $\mathbf{y}$ vector pair is presented to the network the processing elements of layer 3 compete

The author is with Hecht-Nielsen Neurocomputer Corporation, 5893 Oberlin Drive, San Diego, California 92121.

Received 18 July 1987.
0003-6935/87/234979-06$02.00/0.

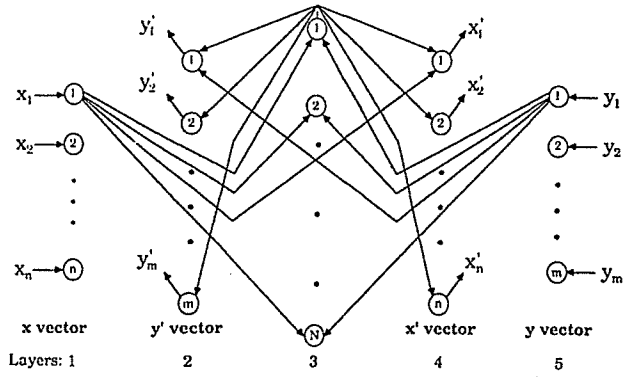

Fig. 1. Topology of the counterpropagation network.

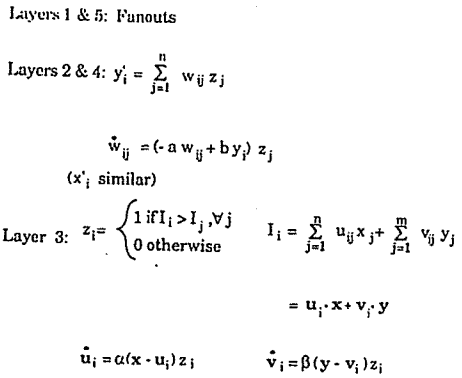

Fig. 2. CPN mathematics.

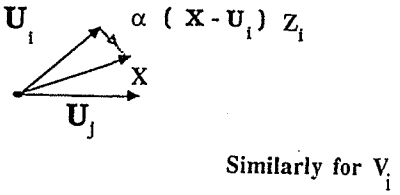

Fig. 3. Kohonen's weight change law.

with one another as shown in the z_i equation of Fig. 2. The processing element with the highest average weight vector correlations with **x** and **y** (i.e., the highest value of I_i) has its output signal (z_i) set to one. All $N-1$ other z_i outputs are set to zero. Ties are broken on the basis of the smallest processing element index. By virtue of the form of the layer 3 weight change law (shown at the bottom of Fig. 2), only the best matching processing element on layer 3 gets to adjust its weight vector in response to each vector pair input during training.

As shown by Kohonen,[2] the ($\mathbf{u}_i$,$\mathbf{v}_i$) weight vectors of layer 3 (which start out as unit vectors) will self-organize in response to the input vector pairs so that these vectors will, after statistical equilibration, have the following two properties:

(1) The vectors $\mathbf{u}_i$ and $\mathbf{v}_i$ will remain approximately unit vectors.

(2) The weight vectors $\mathbf{u}_i$,$\mathbf{v}_i$ will be distributed on the Cartesian product unit sphere in such a way that the probability of a given randomly chosen (**x**,**y**) (the **x** vectors are assumed to be chosen in accordance with a fixed probability density function ρ and this then induces a density for the **y** vectors) being closest to any one of these weight vectors is equal to approximately $1/N$. In other words, the win regions of these weight vectors are equally likely to contain an input vector pair chosen in accordance with ρ (see Sec. III and Ref. 4 for the definition of a win region). [Note: Proposition 5.1 in Chap. 5 of Ref. 2 is not exactly correct (personal communication from T. Kohonen). A density correction factor is required because the weight vectors tend to oversample high density areas and undersample low density areas. A practical cure for this problem is suggested below.]

Figure 3 clarifies this behavior. The weight vectors $\mathbf{u}_i$ and $\mathbf{v}_i$ of the winning processing element on layer 3 are moved toward **x** and **y**, respectively, by an amount equal to the fraction α of the distance between them and their targets. Since these vectors are all normalized, these movements are approximately perpendicular to the directions of $\mathbf{u}_i$ and $\mathbf{v}_i$, and thus do not change their length much. In fact, if either of these vectors gets significantly longer or shorter than unit length, this weight adjustment process will act to restore their length to one by dragging them up or down radially as well as turning them. This process ends up moving the weight vectors of layer 3 around to an equilibrium configuration that represents the average of the effects of a huge number of adjustments. Thus, they become representative of the distribution of the input data vector pairs. Thus, the weight vectors of the processing elements of layer 3 become organized as a more or less statistically optimal set of examples of the relationships between the **x** and **y** vectors. Note that, if the function ϕ is invertible, layer 3 will be equally sensitive to both the forward and backward mappings. Weight changes with layer 3 processing elements take place in accordance with the two equations at the bottom of Fig. 2. This Kohonen learning law rotates the closest matching weight vectors around toward the latest input vectors. This movement is typically almost perpendicualr to the weight vector and thus has the effect of keeping the vector approximately normalized (see Ref. 2 for details on Kohonen learning).

Layers 2 and 4 of the network learn the average **x** and **y** vector values that occur when each of the processing elements of layer 3 wins the closeness competition. Such a structure is called an outstar and was invented by Grossberg.[1] Grossberg has shown that, if the constants a and b in the layer 2 and 4 equations of Fig. 2 are set equal to each other at a positive value less than one, the **x**′ and **y**′ values emitted by layers 2 and 4 will, after statistical equilibrium, equal approximately the average value of the **x** and **y** vectors that historically allowed that particular layer 3 element to win the competition of that layer.

In summary, after both layer 3 and layers 2 and 4 equilibrate, an input vector pair **x** and **y** will cause the network to emit **x**′ and **y**′ vectors that are approxi-

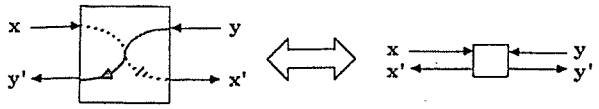

Fig. 4. The CPN module.

mately the same as the nearest matching layer 3 weight vector pair. If the pair $\mathbf{x}$ and $\mathbf{0}$ (i.e., the $\mathbf{y}$ vector is set to zero) are entered, the output pair will again be essentially the same as the layer 3 weight vector pair with a first vector element that best matches $\mathbf{x}$. If ϕ is invertible the same sort of result will hold if a pair $\mathbf{o}$ and $\mathbf{y}$ is entered. If a pair having some components of both the $\mathbf{x}$ and $\mathbf{y}$ vectors zeroed out is entered, the network will complete the vector pair and the outputs will be approximately the same as the best matching layer three weight vector's elements. Thus, CPN functions very much like a lookup table. The overall CPN can be viewed as a modular element that will self-organize to form such a table and then respond to partial inputs as outlined above. Figure 4 shows a schematic representation for this CPN module. This module can be used as a building block for creating more complex and more capable networks.

III. CPN Error Analysis

Since the output of a CPN is almost exactly equal to one of the weight vectors (this is also true for the forward-only CPN version considered below), let us analyze the mean-squared error of a CPN system on this basis. To stay within the confines of conventional notation (and to emphasize the universality of the discussion) the layer three weight vectors are named $\mathbf{w}_i$ even though this contradicts the notation of Sec. II. In fact, several of the variables (e.g., n,$\mathbf{z}$,$\mathbf{v}$) used previously will be reused in this section with different meanings.

The first issue is to define the input/output behavior of the CPN system. The basic idea is that an input vector $\mathbf{z}$ (consisting perhaps of the input vector pair $\mathbf{x}$ and $\mathbf{y}$ of Sec. II) comes into the system. The network then compares $\mathbf{z}$ with all the weight vectors $\mathbf{w}_i$, $i = 1,2,\ldots,N$ of the system and emits the nearest matching vector $\mathbf{w}_k$ as its output. This is the basic function of the CPN. As in the previous section, $\mathbf{z}$ is assumed to be randomly selected in accordance with a fixed probability density function ρ.

Given the above definitions, the mean-squared error $F(W)$ of this system is the average of the square of the distance between an arbitrary input vector $\mathbf{z}$ and the nearest matching $\mathbf{w}_i$, weighted by the probability $\rho(\mathbf{z})$, which is given by

$$F(W) = \int_{\Omega n} \theta(W,\mathbf{z})\rho(\mathbf{z})dA(\mathbf{z}),$$

where

$$W = [\mathbf{w}_1,\mathbf{w}_2,\ldots,\mathbf{w}_N]$$

$$\mathbf{z},\mathbf{w}_i \in \Omega^n \equiv \{\mathbf{x} \in \mathbf{R}^n \mid |\mathbf{x}| = 1\},$$

$$\theta(W,\mathbf{z}) \equiv \min(|\mathbf{z}-\mathbf{w}_1|^2,|\mathbf{z}-\mathbf{w}_2|^2,\ldots,|\mathbf{z}-\mathbf{w}_N|^2).$$

This can be simplified by noting that

$$|\mathbf{z}-\mathbf{w}_i|^2 = |\mathbf{z}|^2 + |\mathbf{w}_i|^2 - 2\mathbf{z}\cdot\mathbf{w}_i,$$

and thus

$$\theta(W,\mathbf{z}) = 2 - 2\max(\mathbf{z}\cdot\mathbf{w}_1,\mathbf{z}\cdot\mathbf{w}_2,\ldots,\mathbf{z}\cdot\mathbf{w}_N).$$

By defining the win region of $\mathbf{w}_i$ to be

$$B_i \equiv \{\mathbf{z} \in \Omega^n \mid i \text{ is the smallest integer } 1 \le i \le N$$

$$\text{such that } \mathbf{z}\cdot\mathbf{w}_i \ge \mathbf{z}\cdot\mathbf{w}_j \forall j\},$$

we can reexpress the integral for $F(W)$ as

$$F(W) = 2 - 2\sum_{i=1}^{N} \mathbf{w}_i \cdot \int_{B_i} \mathbf{z}\rho(\mathbf{z})dA(\mathbf{z}),$$

since for $\mathbf{z} \in B_i$,

$$\mathbf{z}\cdot\mathbf{w}_i = \max(\mathbf{z}\cdot\mathbf{w}_1,\mathbf{z}\cdot\mathbf{w}_2,\ldots,\mathbf{z}\cdot\mathbf{w}_N),$$

and since $\mathbf{w}_i$ is a constant vector. Note that if m_i is defined to be

$$m_i \equiv \int_{B_i} \rho(\mathbf{z})dA(\mathbf{z})$$

and $\mathbf{v}_i$ (the centroid of the sphere subset B_i) is defined to be

$$\mathbf{v}_i \equiv (1/m_i)\int_{B_i} \mathbf{z}\rho(\mathbf{z})dA(\mathbf{z})$$

we can rewrite $F(W)$ as

$$F(W) = 2 - 2\sum_{i=1}^{N} m_i(\mathbf{w}_i\cdot\mathbf{v}_i).$$

Note that

$$\sum_{i=1}^{N} m_i = 1.$$

The geometry of $F(W)$ is that it gets smaller as the B_i regions get smaller and more uniform in importance (as determined by m_i). Note that $\mathbf{v}_i$ can be thought of as the average vector in B_i. If B_i is a small, localized, almost planar region, $\mathbf{v}_i$ will be almost a unit vector that points to the probability-weighted center of B_i. This will make the dot product $\mathbf{w}_i\cdot\mathbf{v}_i$ almost one, since $\mathbf{w}_i$ is a unit vector that also points at the center of B_i. However, if B_i is very large and significantly curved, $\mathbf{v}_i$ will be significantly shorter than a unit vector and the dot product $\mathbf{w}_i\cdot\mathbf{v}_i$ will be smaller, thus increasing $F(W)$. Finally, note that the sum of the $\mathbf{v}_i$ vectors is a constant vector that only depends on ρ and not on the weight vectors:

$$\sum_{i=1}^{N} \mathbf{v}_i = \int_{\Omega n} \mathbf{z}\rho(\mathbf{z})dA(\mathbf{z}).$$

In conclusion, a relatively simple formula exists for the mean-squared error $F(W)$ of the CPN. By virtue of the generality of the approach taken in this definition this formula applies to both the full CPN and the

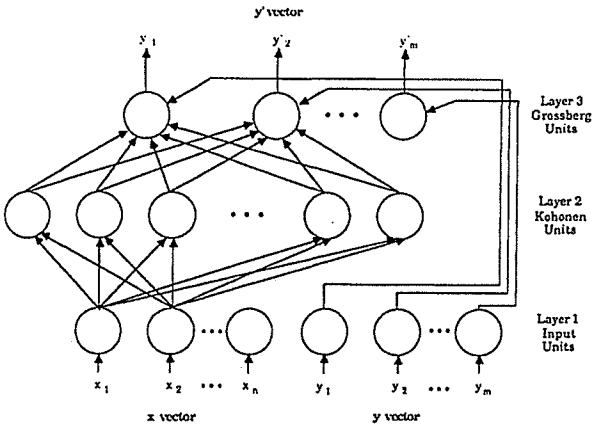

Fig. 5. Forward-only CPN.

forward-only version defined below. As is clear from the form of the equation, the dependence of the error on each coordinate of the input vectors is uniform across all the coordinates. Thus, no matter how important or unimportant a coordinate is for the implementation of a mapping its contribution to the adjustment of the Kohonen units is approximately the same (this is particularly so for the foward-only version of the network). This effect fundamentally limits the accuracy of any lookup table approximation.

IV. CPN Variants and Evolutes

Multiple types of counterpropagation network have been defined; with the CPN module defined above being the most imporant example. One useful CPN variant is illustrated in Fig. 5. In this network layer 4 of the basic CPN has been excised, as have the interconnects from the y input vector to layer 3 (this last change is optional). The net result is a CPN variant (called the forward-only CPN module) that is ideal for problems in which only transformations from x to y are of interest (and implementation of the inverse mapping and completions of partial x–y input pairs are not of interest).

Finally, by modifying the competition process on layer 3 of the CPN, there can be K winning processing elements (coresponding to the K nearest matching $\mathbf{w}_i$ vectors) instead of one (see Ref. 4 for additional discussion about the advantages of this). If the outputs of these K processing elements are set so that they sum to one (i.e., the former single output signal of one is now divided among the K units in some manner directly dependent on their relative I_i values) the outstars of layers 2 and 4 will blend their usual output values. This interpolation process can lead to significantly increased mapping approximation accuracy for no increase in network size. A CPN using this approach is operating in an interpolative mode, as opposed to the accretive mode discussed above.

V. Discussion

One of the problems with CPN (returning now to the notation of Figs. 1 and 2) is that Kohonen learning only works well when ρ is nonzero only in a single connected region. If this is not true, weight vectors can get stuck in isolated regions and cannot move to where they are needed. A number of solutions have been developed for this. One is called convex combination. In this approach, one starts off with all the (unit) weight vectors equal to the vector $\mathbf{1} = (1/\sqrt{n}, 1/\sqrt{n}, \ldots, 1/\sqrt{n})$. The x–y data inputs then start out as normalized convex combinations [i.e.,

$$\widehat{\alpha\mathbf{x} + (1-\alpha)\mathbf{1}}]$$

of their actual values and this same vector. Initially, the combinations start off with a very low value of α (near zero). This forces all the data vectors to be close to the weight vectors. As time goes on the value of α is raised slowly to one. As this happens, the weight vectors are peeled off and follow the input data vectors as they move away from 1. This works very well but it slows down adaptation. Another approach is to add noise to the data, which has the effect of making ρ positive everywhere. This works all right but is much slower than convex combination. Finally, another approach is to build a conscience into each Kohonen processing element to monitor its success in the layer 3 competition. If it wins the competition substantially more often than $1/N$ of the time, have that unit take itself out of the competition for a while, allowing stuck units to win their way out of captivity. Initial work suggests that this approach works well and may solve the problems with Kohonen's Proposition 5.1. This idea of adding a conscience to the layer 3 processing elements is due to Duane DeSieno of Logical Designs in San Diego.

Given its simple lookup table function, CPN is obviously inferior to backpropagation for most mapping network applications. Its advantages are that it is simple and that it forms a good statistical model of its input vector environment (i.e., of ρ). Another potential advantage is the existence of a closed-form formula for the mean-squared error of the network's mapping function. As with all lookup table approaches, CPN requires a number of entries (Kohonen units) to achieve high mapping approximation accuracy. However, because CPN adjusts its weights to conform to the statistics of the input vectors the number of Kohonen processing elements required to reach a particular level of mapping approximation accuracy depends primarily on the complexity of the mapping and the statistics of the x and y vector selections, and not very much on the dimensionality of the spaces involved. Thus, CPN avoids the worst part of Richard Bellman's "curse of high dimensionality." The use of CPN in an interpolation mode can also help accuracy considerably. CPN seems to be well suited to situations where the relationships between the input data components are more important than the implementation of a mapping (although it does fairly well as a mapping network). The network is also useful for rapid prototyping of neurocomputing systems (even if a backpropagation or some other mapping network will be used in its place late after the rest of the system is working)

because it typically converges 1 or 2 orders of magnitude faster than backpropagation. The CPN module can be used during early development and can later be replaced by a plug-compatible network such as backpropagation.

Finally, CPN illustrates a key point about neurocomputing system design, namely, that many of the existing network paradigms can be viewed as building block components that can be assembled into new configurations that offer new information processing capabilities.

References

1. D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vols. 1, 2, 3* (MIT Press, Cambridge, MA, 1986, 1987).
2. T. Kohonen, *Self-Organization and Associative Memory* (Springer-Verlag, New York, 1984).
3. S. Grossberg, *Studies of Mind and Brain* (Reidel, Boston, MA, 1982).
4. R. Hecht-Nielsen, "Combinatorial Hypercompression," in *Proceedings, IEEE International Conference on Neural Networks* (IEEE, New York, 1987).

Bibliography

S. Amari, "Field Theory of Self-Organizing Neural Networks," IEEE Trans. Syst. Man Cybern. **SMC-13**, 741 (1983).

S. Amari and M. A. Arbib, *Competition and Cooperation in Neural Nets* (Springer-Verlag, New York, 1982).

S. Amari, "A Method of Statistical Neurodynamics," Biol. Cybern. **14**, 201 (1974).

S. Amari, "Characteristics of Randomly Connected Threshold-Element Networks and Network Systems," Proc. IEEE **59**, 35 (1971).

J. A. Anderson, "Cognitive and Psychological Computation with Neural Models," IEEE Trans. Syst. Man Cybern. **SMC-13**, 799, (1983).

J. A. Anderson, "A Simple Neural Network Generating an Interactive Memory," Math Biosci. **14**, 197 (1972).

A. G. Barto and R. S. Sutton, "Stimulation Experiments with Goal-Seeking Adaptive Elements," AFWAL-TR-84-1022, DTIC Doc. ADA 140295, (Feb. 1984).

A. G. Barto and R. S. Sutton, "Landmark Learning: an Illustration of Associative Search," Biol. Cybern. **42**, 1 (1981). ADA 140295, (Feb. 1984).

A. G. Barto and R. S. Sutton, "Landmark Learning: an illustration of Associative Search," Biol. Cybern. **42**, 1 (1981).

G. A. Carpenter and S. Grossberg, "A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine," Comput. Vision Graphics Image Process. **37**, 54 (1987).

D. Casasent, "Scene Analysis Research: Optical Pattern Recognition and Artificial Intelligence," Proc. Soc. Photo-Opt. Instrum. Eng. **634**, 439 (1986).

M. A. Cohen and S. Grossberg, "Neural Dynamics of Speech and Language Coding: Developmental Programs, Perceptual Grouping, and Competition for Short Term Memory," Hum. Neurobiol. **5**, 1 (1986).

M. A. Cohen and S. Grossberg, "Absolute Stability of Global Pattern Formation and Parallel Memory Storage by Competitive Neural Networks," IEEE Trans. Syst. Man and Cybern. **SMC-13**, (1983).

T. M. Cover and P. E. Hart, "Nearest Neighbor Pattern Classification," IEEE Trans. Inf. Theory **IT-13**, (1967).

C. Cruz and J. Y. Tam, "NEP: an Emulation-Assist Processor for Parallel Associative Networks," IBM Palo Alto Scientific Center Report G320-3475 (1985).

C. Cruz and H. J. Myers, "Associative Networks II," IBM Palo Alto Scientific Center Report G320-3446 (1983).

C. Cruz and H. J. Myers, "Associative Networks," IBM Palo Alto Scientific Center Report G320-3474 (1982).

J. G. Daugman, "Uncertainty Relation for Resolution in Space, Spatial Frequency, and Orientation Optimized by Two-Dimensional Visual Cortical Filters," J. Opt. Soc. Am. A **2**, 1160 (1985).

J. Denker, *Proc. Second Annual Conference on Neural Networks for Computing*, AIP Conf. Proc. **151**, American Institute of Physics, New York, (1986).

G. J. Dunning, E. Marom, Y. Owechko, and B. H. Soffer, "Optical Holographic Associative Memory Using a Phase Conjugate Resonator," Proc. Soc. Photo-Opt. Instrum. Eng. **625**, 205 (1986).

S. J. Farlow, Ed., *Self-Organizing Methods in Modeling: GMDH Type Algorithms* (Marcel Dekker, New York, 1984).

A. D. Fisher, R. C. Fukuda, and J. N. Lee, "Implementations of Adaptive Associative Optical Computing Elements," Proc. Soc. Photo-Opt. Instrum. Eng. **625**, 1 (1986).

A. D. Fisher and C. L. Giles, "Optical Adaptive Associative Computer Architectures," in *Proceedings, IEEE COMPCOM Meeting*, IEEE Catalog CH2135-2/85 (Feb. 1985), pp. 342–344.

A. D. Fisher, C. L. Giles, and J. N. Lee, "Associative Processor Architectures for Optical Computing," J. Opt. Soc. Am. A **1**, 1337 (1984).

K. Fukushima and S. Miyake, "Neocognition: a New Algorithm for Pattern Recognition Tolerant of Deformations and Shifts in Position," Pattern Recognition **15**, 455 (1984).

S. Geman, "The Law of Large Numbers in Neural Modeling," in *Mathematical Psychology and Psychophysiology* S. Grossberg, Ed. (American Mathematical Society, Providence, RI, 1981).

S. Grossberg and G. Stone, "Neural Dynamics of Word Recognition and Recall; Attentional Printing, Learning, and Resonance," Psychol. Rev. **93**, 46 (1986).

S. Grossberg and M. Kuperstein, *Neural Dynamics of Adaptive Sensory-Motor Control* (North-Holland, Amsterdam, 1986).

S. Grossberg and E. Mingolla, "Neural Dynamics of Perceptual Grouping: Textures, Boundaries, and Emergent Segmentations," Percept. Psychophys. **38**, 141 (1985).

S. Grossberg and M. A. Cohen, "Some Global Properties of Binocular Resonances: Disparity Matching, Filling-In, and Figure-Ground Synthesis," in *Figural Synthesis*, T. Caelli and P. Dodwell, Eds. (Erlbaum, Hillsdale, NJ. 1984).

S. Grossberg, "Embedding Fields: Underlying Philosophy, Mathematics, and Applications to Psychology, Phsiology, and Anatomy," J. Cybern. **1**, 28 (1971).

S. Grossberg, "Some Networks That Can Learn, Remember, and Reproduce Any Number of Complicated Space-Time Patterns, II," Stud. Appl. Math. **49**, 135 (1970).

S. Grossberg, "Embedding Fields: a Theory of Learning with Physiological Implications," J. Math Psychol. **6**, 209 (1969).

S. Grossberg, "Some Networks That Can Learn, Remember, and Reproduce Any Number of Complicated Space-Time Patterns, I," J. Math Mech. **19**, 53 (1969).

S. Grossberg, "Nonlinear Difference-Differential Equations in Prediction and Learning Theory," Proc. Natl. Acad. Sci. U.S.A. **58**, 1329 (1967).

R. Hecht-Nielsen, "Kolmogorov's Mapping Neural Network Existence Theorem," in *Proceedings, IEEE International Conference on Neural Networks* (IEEE, New York, 1987).

R. Hecht-Nielsen, "Counterpropagation Networks," in *Proceedings, IEEE International Conference on Neural Networks* (IEEE, New York, 1987).

R. Hecht-Nielsen, "Neurocomputer Applications," in *Proceedings, National Computer Conference 1987* (American Federation of Information Processing Societies, Reston, VA 1987).

R. Hecht-Nielsen, "Neural Network Nearest Matched Filter Classification of Spatiotemporal Patterns," Appl. Opt. **26**, 1892 (1987).

R. Hecht-Nielsen, "Performance Limits of Optical, Electro-Optical,

and Electronic Artificial Neural System Processors," Proc. Soc. Photo-Opt. Instrum. Eng. 634, 277 (1986).
R. Hecht-Nielsen, "Book Review of Grossberg's *STUDIES OF MIND AND BRAIN*," J. Math. Psychol. 27, 335 (1983).
R. Hecht-Nielsen, "Neural Analog Processing," Proc. Soc. Photo-Opt. Instrum. Eng. 360, 180 (1982).
R. Hecht-Nielsen, "Neural Analog Information Processing," Proc. Soc. Photo-Opt. Instrum. Eng. 298, 138 (1981).
G. E. Hinton and J. A. Anderson, Eds., *Parallel Models of Associative Memory* (Erlbaum, Hillsdale, NJ, 1981).
J. J. Hopfield and D. W. Tank, "Neural Computation of Decisions in Optimization Problems," Biol. Cybern. 52, 141 (1985).
J. J. Hopfield, "Neurons with Graded Response Have Collective Computational Properties Like Those of Two-State Neurons," Proc. Natl. Acad. Sci. U.S.A. 81, (1984).
J. J. Hopfield, "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," Proc. Natl. Acad. Sci. U.S.A. 79, 2254 (1982).
A. H. Klopf, *The Hedonistic Neuron* (Hemisphere, Washington, DC, 1982).
A. H. Klopf and E. Gose, "An Evolutionary Pattern Recognition Network," IEEE Trans. Syst. Man Cybern. SMC-5, 247 (1969).
B. Kosko, "Bidirectional Associative Memories," to be published.
B. Kosko, "Fuzzy Associative Memories," in *Fuzzy Expert Systems*, A. Kandel, Ed. (Addison-Wesley, Reading, MA, 1987).
B. Kosko, "Fuzzy Entropy and Conditioning," Inf. Sci. 40, 165 (1986).
B. Kosko, "Fuzzy Knowledge Combination," Int. J. Intell. Syst. 1, 293 (1986).
B. Kosko and J. Limm, "Vision as Causal Activation and Association," Proc. Soc. Photo-Opt. Instrum. Eng. 579, 104 (1985).
J. L. McClelland and D. E. Rumelhart, "Distributed Memory and the Representation of General and Specific Information," J. Exp. Psychol. 114, 159 (1985).
R. J. McEliece, E. C. Posner, E. R. Rodemich, and S. S. Venkatesh, "The Capacity of the Hopfield Associative Memory," California Institute of Technology (1986).
R. L. Mitchell, "Helicopter Blade Modulation Model (Revised)," Defense Technical Information Center, DTIC AD 1089574 (29 Mar. 1978).
G. Palm, "On Associative Memory," Biol. Cybern. 36, 19 (1980).
D. Psaltis and S. Venkatesh, "Information Storage and Retrieval in Two Associative Nets," California Institute of Technology (1985).
D. Psaltis and Y. S. Abu-Mostafa, "Computation Power of Parallelism in Optical Computers," California Institute of Technology (1985).
D. Psaltis and N. Farhat, "Optical Information Processing Based on an Associative-Memory Model of Neural Nets with Thresholding and Feedback," Opt. Lett. 10, 98 (1985).
D. E. Rumelhart, G. E. Hinton, and R. J. Williams "Learning Internal Representation by Error Propagation," Institute for Cognitive Science Report 8506, UCSD (Sept. 1985).
D. E. Rumelhart and D. Zipser, "Feature Discovery by Competitive Learning," Cognitive Sci. 9, 75 (1985).
T. J. Sejnowski and C. R. Rosenberg, "NETalk: A Parallel Network That Learns to Read Aloud," Johns Hopkins University (Jan. 1986).
T. J. Sejnowski, "Skeleton Filters in the Brain," in *Parallel Models of Associative Memory*, G. E. Hinton and J. A. Anderson, Eds. (Erlbaum, Hillsdale, NJ, 1981).
B. H. Soffer, G. J. Dunning, Y. Owechko, and E. Marom, "Associative Holographic Memory with Feedback Using Phase-Conjugate Mirrors," Opt. Lett. 11, 118 (1986).
K. Steinbuch and U. A. W. Piske, "Learning Matrices and Their Applications," IEEE Trans. Electron. Comput. EC-12, 846 (1963).
K. Steinbuch, "Die Lernmatrix," Kybernetik 1, 36 (1961).
A. Thakoor, "Content-Addressable, High Density Memories Based on Neural Network Models," JPL Report D-4166 (Mar. 1987).
B. Widrow and D. Stearns, *Adaptive Signal Processing* (Prentice-Hall, Englewood Cliffs, NJ, 1985).
B. Widrow, "Generalization Information Storage in Networks of ADALINE Neurons," in *Self-Organizing Systems*, G. T. Yovitts, Ed. (Spartan, Washington, DC 1962).
B. Widrow and M. Hoff, Jr., "Adaptive Switching Circuits," IRE WESCON Conv. Record, Part 4, 96 (1960).
D. J. Willshaw, "Models of Distributed Associative Memory," Ph.D. Thesis, U. Edinburgh (1971).
D. J. Willshaw, and H. C. Longuet-Higgins, "Associative Memory Models," in *Machine Intelligence*, B. Meltzer and O. Michie, Eds. (Edinburgh U. P., 1970).
D. J. Willshaw, O. P. Buneman, and H. C. Longuet-Higgins, "Non-Holographic Associative Memory," Nature London 222, 960 (1969).
P. M. Woodward, *Probability and Information Theory with Applications to Radar* (Pergamon, New York, 1953).