Pergamon

Neural Networks, Vol. 7, No 1, pp 159–168, 1994

0893-6080/94 $6.00 + .00

CONTRIBUTED ARTICLE

A Distributed Outstar Network for Spatial Pattern Learning

GAIL A. CARPENTER

Boston University

(*Received* 23 *March* 1993; *accepted* 12 *July* 1993)

Abstract—*The distributed outstar, a generalization of the outstar neural network for spatial pattern learning, is introduced In the outstar, signals from a source node cause weights to learn and recall arbitrary patterns across a target field of nodes The distributed outstar replaces the outstar source node with a source field of arbitrarily many nodes, whose activity pattern may be arbitrarily distributed or compressed Learning proceeds according to a principle of atrophy due to disuse, whereby a path weight decreases in joint proportion to the transmitted path signal and the degree of disuse of the target node. During learning, the total signal to a target node converges toward that node's activity level Weight changes at a node are apportioned according to the distributed pattern of converging signals Three synaptic transmission functions, a product rule, a capacity rule, and a threshold rule, are examined for this system. The three rules are computationally equivalent when source field activity is maximally compressed, or winner-take-all. When source field activity is distributed, catastrophic forgetting may occur Only the threshold rule solves this problem. Analysis of spatial pattern learning by distributed codes thereby leads to the conjecture that the unit of long-term memory in such a system is an adaptive threshold, rather than the multiplicative path weight widely used in neural models*

Keywords—Spatial pattern learning, Distributed code, Outstar, Adaptive threshold, Rectified bias, Atrophy due to disuse, Transmission function, Neural network.

1. INTRODUCTION: OUTSTAR LEARNING AND DISTRIBUTED CODES

An *outstar* is a neural network that can learn and recall arbitrary spatial patterns (Grossberg, 1968a). Outstar learning and recall occur when a source node transmits a weighted signal to a target, or border, field of nodes. This network is a key component of various neural models of cognitive processing. For example, the outstar has been identified as a minimal neural network capable of classical conditioning (Grossberg, 1968b, 1974). In terms of stimulus sampling theory (Estes, 1955), the source node plays the role of a sampling cell. When the sampling cell is active, long-term memory (LTM) traces, or adaptive weights, learn stimulus sampling probabilities of border field activity patterns. A sequence of outstars, called an *avalanche*, forms a minimal network capable of learning and ritualistic performance of an arbitrary space-time pattern (Grossberg, 1969). Within the adaptive resonance theory of self-organizing pattern classification, outstars learn the top-down expectations that are critical to code stabilization (Grossberg, 1976). All neural network realizations of adaptive resonance theory (ART models) have so far used outstar learning in the top-down adaptive filter (Carpenter & Grossberg, 1987a,b, 1990; Carpenter, Grossberg, & Rosen, 1991a). The supervised ARTMAP system (Carpenter, Grossberg, & Reynolds, 1991) also employs outstar learning in the formation of its predictive maps. Outstars have thus played a central role in both the theoretical analysis of cognitive phenomena and the neural models that realize the theories, as well as in applications of these systems.

An outstar is characterized by one source node sending weighted inputs to a target field. We will here consider spatial pattern learning in a more general setting, in which an arbitrarily large source field replaces the single source node of the outstar. This *distributed outstar network* (Figure 1) reduces to the original outstar when the source field F_2 consists of a single node. Then, weights in the $F_2 \rightarrow F_1$ adaptive filter track the F_1 activity pattern when the one F_2 node is active.

At first, distributed outstar learning would appear to be modeled already in the ART top-down adaptive

Acknowledgements: This research was supported in part by British Petroleum (BP 89-A-1204), ARPA (ONR N00014-92-J-4015), the National Science Foundation (NSF IRI-90-00530), and the Office of Naval Research (ONR N00014-91-J-4100). The author wishes to thank Diana J. Meyers for her valuable assistance in the preparation of the manuscript.

Requests for reprints should be sent to the author at Center for Adaptive Systems and Department of Cognitive and Neural Systems, Boston University, 111 Cummington Street, Boston, MA 02215.

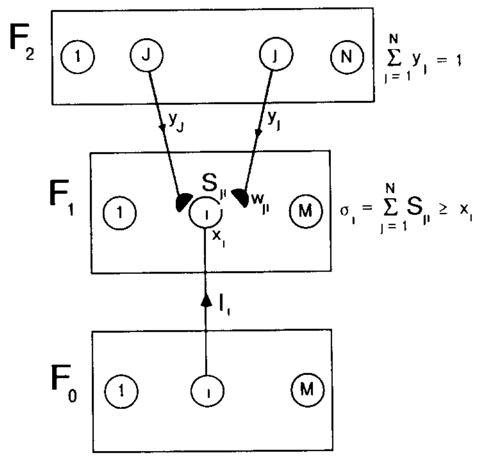

FIGURE 1. Distributed outstar network for spatial pattern learning. During adaptation a top-down weight w_{ji}, from the jth node of the coding field F_2 to the ith node of the pattern registration field F_1, may decrease or remain constant. An atrophy due to disuse learning law causes the total signal σ_i from F_2 to the ith F_1 node to decay toward that node's activity level x_i, if σ_i is initially greater than x_i. Within this context, three synaptic transmission rules are analyzed.

filter. However, to date, networks that explicitly realize adaptive resonance assume the special case in which F_2 is a *choice,* or *winner-take-all,* network. In this case, only one F_2 node is active during learning, so each F_2 node acts, in turn, as an outstar source node. We will consider how to design a spatial pattern learning network that allows the activity pattern at the coding field F_2 to be arbitrarily distributed (Section 2). That is, one, several, or all of the F_2 nodes may be active during learning.

One possible design is simply to implement outstar learning in each active path. However, such a system is subject to catastrophic forgetting that can quickly render the network useless, unless learning rates are very slow (Section 3). In particular, if all F_2 nodes were active during learning, all $F_2 \twoheadrightarrow F_1$ weight vectors would converge toward a common pattern.

A learning principle of *atrophy due to disuse* leads toward a solution of the catastrophic forgetting problem (Section 4). By this principle, a weight in an active path is assumed to atrophy, or decay, in joint proportion to the size of the transmitted synaptic signal and a suitably defined *degree of disuse* of the target cell. During learning, the total transmitted signal from F_2 converges toward the activity level of the target F_1 node. Atrophy due to disuse thereby dynamically substitutes the total $F_2 \twoheadrightarrow F_1$ signal for the individual outstar weight. This seems a plausible step toward spatial pattern learning by a coding source field instead of by a single source node. Unfortunately, this development is, by itself, insufficient. In particular, the network still suffers catastrophic forgetting if signal transmission obeys a *product rule* This rule, now used in nearly all neural models, assumes that the transmitted synaptic signal from the jth F_2 node to the ith F_1 node is proportional to the product of the path signal y_j and the path weight w_{ji}. An alternative transmission process, one that has been used in a neural network realization of fuzzy ART (Carpenter, Grossberg, & Rosen, 1991b; Carpenter & Grossberg, 1993), is described by a *capacity rule* (Section 5). However, catastrophic forgetting is even more serious a problem for this rule than for the product rule.

Fortunately, another plausible synaptic transmission rule solves the problem (Section 6). This *threshold rule* postulates a transmitted signal equal to the amount by which the $F_2 \twoheadrightarrow F_1$ signal y_j exceeds an adaptive threshold τ_{ji}. Where weights decrease during atrophy due to disuse learning thresholds increase· formally, τ_{ji} is identified with $(1 - w_{ji})$. When synaptic transmission is implemented by a threshold rule, weight/threshold changes are bounded and automatically apportioned according to the distribution of F_2 activity, with fast learning as well as slow learning. When F_2 makes a choice, the three synaptic transmission rules are computationally identical, and atrophy due to disuse learning is essentially the same as outstar learning. Thus, functional differences between the three types of transmission would be experimentally and computationally measurable only in situations where the F_2 code is distributed.

Computational analysis of distributed codes hereby leads unexpectedly to a hypothesis about the mechanism of synaptic transmission in spatial pattern learning systems. That is, the unit of long-term memory in these systems is conjectured to be an adaptive threshold, rather than a multiplicative path weight. Historically, early definitions of the perceptron specified a general class of synaptic transmission rules (Rosenblatt, 1958, 1962). However, the electrical switching circuit model, which realizes multiplicative weights as adjustable gains, quickly became the dominant metaphor (Widrow & Hoff, 1960). Over the ensuing decades, efficient integrated hardware realization of the linear adaptive filter has remained a challenge. In opto-electronic neural networks, the adaptive threshold synaptic transmission rule, realized as a rectified bias, may be easier to implement than on-line multiplication (T. Caudell, personal communication). Thus, even in networks where the product rule and the threshold rule are computationally equivalent, their diverging physical interpretations may prove significant, in both the neural and the hardware domains.

The adaptive threshold hypothesis leads to the *distributed outstar learning law,* summarized in Section 7. Section 8 concludes with an example that illustrates distributed outstar dynamics by means of a network that has two nodes in the source field.

2. SPATIAL PATTERN LEARNING

The distributed outstar network (Figure 1) features an adaptive filter from a *coding field* F_2 to a *pattern reg-*

istration field F_1. The role of this filter is to carry out spatial pattern learning, whereby the adaptive path weights track the activity pattern of the target field, F_1. When F_2 consists of just one node ($N = 1$) the network reduces to the outstar. During outstar learning, weights in the paths emanating from an F_2 node track F_1 activity. That is, when the jth F_2 node is active, the weight vector $\mathbf{w}_j \equiv (w_{j1}, \ldots w_{ji}, \ldots w_{jM})$ converges toward the F_1 activity vector $\mathbf{x} \equiv (x_1, \ldots x_i, \ldots x_M)$ of the target, or border, nodes at the outer fringe of the filter.

Although many variants of outstar learning have been analyzed (Grossberg, 1968a, 1972), the essential outstar dynamics are described by the equation:

Basic outstar

$$\frac{d}{dt} w_{ji} = y_j(x_i - w_{ji}) \tag{1}$$

This is the learning law used, for example, in the top-down adaptive filters of ART 1 (Carpenter & Grossberg, 1987a), ART 2 (Carpenter & Grossberg, 1987b), and fuzzy ART (Carpenter et al., 1991a). By eqn (1), $w_{ji} \rightarrow x_i$ when $y_j > 0$. When $y_j = 0$, w_{ji} remains constant. The term $y_j x_i$ in eqn (1) describes a Hebbian correlation whereby the weight tends to increase when both the presynaptic F_2 node j and the postsynaptic F_1 node i are active. The term $-y_j w_{ji}$ describes an anti-Hebbian process whereby the weight w_{ji} tends to decrease when the presynaptic node j is active but the postsynaptic node i is inactive (pre- without post-).

Note that the distributed outstar network in Figure 1 does not constitute a stand-alone pattern recognition system. Typically, this module would be embedded within a larger neural network architecture for supervised or unsupervised pattern learning and recognition. For example, in an ART system the top-down $F_2 \rightarrow F_1$ filter plays a crucial role in ART code stabilization. However, additional network elements are needed to determine which F_2 code will be selected by an input $\mathbf{I}$ in the first place, as well as to implement search and other mechanisms of internal dynamic control (Carpenter & Grossberg, 1987a). We will focus only on design issues pertaining to the top-down adaptive filter.

3. CATASTROPHIC FORGETTING

The distributed outstar network for spatial pattern learning (Figure 1) needs to be designed in such a way as to solve a potential catastrophic forgetting problem. Suppose, for example, that all F_2 nodes are active ($y_j > 0$) at some time when the ith F_1 node is inactive ($x_i = 0$) due, say, to the fact that there is no input to that node at that time ($I_i = 0$). With fast learning, an outstar (1) would send all weights w_{ji} ($j = 1, \ldots, N$) to 0. Within an ART system, general stability requirements would imply that these weights then remain 0 forever. Moreover, no future input I_i to the ith F_1 node could even activate that node, once F_2 became active. If similar weight decays occurred at each F_1 node, all weights would decay to 0. The network would thus quickly become useless, quenching all F_1 activity as soon as any F_2 code was selected.

The special class of F_2 networks called choice, or winner-take-all, systems sidesteps this catastrophic forgetting problem. A code representation field F_2 is a choice network when internal competitive dynamics concentrate all activity at one node (Grossberg, 1973). An F_2 code that chooses the Jth node is described by:

$\mathbf{F_2}$ choice

$$y_j = \begin{cases} 1 & \text{if} \quad j = J \\ 0 & \text{if} \quad j \neq J. \end{cases} \tag{2}$$

In this case, each F_2 node may then be identified with a class, or category, of inputs $\mathbf{I}$. Outstar learning (1) permits a weight w_{ji} to change only if the jth F_2 node is active. When F_2 chooses the node J, all other nodes ($j \neq J$) are inactive. Thus, only the weight w_{Ji} tracks activity at the ith F_1 node:

$$\mathbf{w}_J \rightarrow \mathbf{x}. \tag{3}$$

Even if w_{Ji} decays to 0, all other weights to the ith F_1 node remain unchanged when the Jth category is selected. These other weights are thus able to learn their own F_1 patterns when they later become active.

Choice represents an extreme form of STM competition at F_2. By confining all weight changes to a single category, F_2 choice protects the learned codes of all the other categories during outstar learning. However, outstar learning poses a problem when F_2 category representations can be distributed. If a code $\mathbf{y}$ were highly distributed, with all $y_j > 0$, then the outstar learning law (1) would imply that all weight vectors $\mathbf{w}_j$ would converge toward the same F_1 activity vector $\mathbf{x}$. The size of y_j would affect the rate of convergence, but not the asymptotic state of the weights. The severity of this problem can be reduced if learning intervals are required to be extremely short. Then, because the rate at which $\mathbf{w}_j$ approaches $\mathbf{x}$ is proportional to y_j, little change will occur in weights w_{ji} with small y_j. If, however, many of the y_j values are nearly uniform or if learning is not always slow, catastrophic forgetting will occur as all weight vectors approach one common pattern, independently of all their prior learned differences.

A new adaptation rule, called the distributed outstar learning law, solves this problem. Even with fast learning, where weights approach asymptote on each input presentation, the distributed outstar apportions weight changes across active paths without catastrophic forgetting. In the distributed outstar, the rate constant for an individual weight w_{ji} becomes an increasing function both of y_j, as in eqn (1), and of w_{ji} itself. When w_{ji} becomes too small, further change is disallowed. Weights, initially large, can only decrease monotonically during learning. Small weights can decrease further only when y_j is close to 1, which occurs when most of the

F_2 activity is concentrated at node j. When F_2 activity is highly distributed only large weights, close to their initial values, are able to change. Moreover, for highly distributed codes, the maximum possible weight change in any single path is small.

The distributed outstar is derived from the notion that the sum of all $F_2 \rightarrow F_1$ transmitted signals, rather than individual path weights, track target node activity during learning. Weight changes are governed by a principle of atrophy due to disuse, as described in the next section. Within this context, three signal transmission rules are examined (Section 5). An adaptive threshold rule for synaptic transmission is more computationally successful than either of the other two rules, as shown in Section 6.

4. LEARNING BY ATROPHY DUE TO DISUSE

The principle of atrophy due to disuse postulates that the strength of an active path will decay when the path is disused. Active dis-use is distinct from passive non-use, where the strength of an inactive path remains constant, as in eqn (1). To define disuse, a specific class of target fields F_1 will now be considered. So far, no assumptions about the F_1 activity vector **x** have been made. The main hypothesis on F_1 will be that, when F_2 is active, the total top-down input from F_2 to F_1 imposes an upper bound, or limit, on the maximum activity at an F_1 node. In addition to a bottom-up input I_i, a top-down *priming* input from F_2 is assumed to be necessary for an F_1 node to remain active, once F_2 becomes active. This hypothesis is realized by:

Top-down prime

$$0 \le x_i \le \sigma_i, \tag{4}$$

where σ_i is the sum of all transmitted signals S_{ji} from F_2 to the ith F_1 node:

$$\sigma_i = \sum_{j=1}^{N} S_{ji}. \tag{5}$$

In particular, when F_2 is active but $\sigma_i = 0$, no activity can be registered at the ith F_1 node, for any bottom-up input $I_i \in [0, 1]$.

The top-down prime eqn (4) is closely related to the 2/3 Rule of ART (Carpenter & Grossberg, 1987a), which implies that the ith F_1 node will be inactive ($x_i = 0$) if either the bottom-up input I_i is small or the total top-down input σ_i is small when F_2 is active. The 2/3 Rule was derived both from an analysis of system requirements for input registration, priming, and stable, self-organizing pattern learning and classification and from an analysis of the corresponding cognitive phenomena. In binary ART 1 systems with choice at F_2, the 2/3 Rule is realized by allowing the ith F_1 node to be active, when the Jth F_2 node is active, only if $I_i = 1$ and ∂_i exceeds a criterion level, where:

$$\sigma_i = y_J w_{Ji}. \tag{6}$$

Fuzzy ART (Carpenter et al., 1991a), an analog extension of ART 1, realizes the 2/3 Rule by setting:

$$x_i = I_i \wedge w_{Ji} \equiv \min(I_i, w_{Ji}) \tag{7}$$

when the Jth F_2 node is chosen. The symbol $\wedge$ in eqn (7) denotes the fuzzy AND, or intersection, operator. By eqns (2) and (6), when F_2 makes a choice,

$$\sigma_i = w_{Ji}. \tag{8}$$

Equations (7) and (8) suggest setting:

$$x_i = I_i \wedge \sigma_i \tag{9}$$

to define one class of F_1 systems that realize σ_i as a top-down prime, or upper bound, on target node activity x_i.

When F_2 primes F_1, by eqn (4), the *degree of disuse* D_i of the ith F_1 node is defined to be:

$$D_i = (\sigma_i - x_i) \ge 0. \tag{10}$$

When eqn (9) holds,

$$D_i = (\sigma_i - I_i \wedge \sigma_i) = \begin{cases} \sigma_i - I_i & \text{if} \quad \sigma_i \ge I_i \\ 0 & \text{if} \quad \sigma_i \le I_i \end{cases}$$
$$= [\sigma_i - I_i]^+, \tag{11}$$

where

$$[\theta]^+ \equiv \theta \vee 0 \equiv \max(\theta, 0) \tag{12}$$

denotes the rectification operator. In this case, the degree of disuse at the ith F_1 node is the amount by which the top-down input σ_i exceeds the bottom-up input I_i at that node. A learning principle of atrophy due to disuse postulates that a path weight decays in proportion to the degree of disuse of its target node. We here consider a class of learning equations that realize this principle in the form:

$$\frac{d}{dt} w_{ji} = -S_{ji} D_i. \tag{13}$$

Weights can then decay or stay constant, but never grow, when $S_{ji} \ge 0$ and $D_i \ge 0$. With the degree of disuse D_i defined by eqn (10), the learning law (13) becomes:

Atrophy due to disuse

$$\frac{d}{dt} w_{ji} = -S_{ji}(\sigma_i - x_i). \tag{14}$$

In Section 5 three synaptic transmission rules will each define S_{ji} as a function of y_j and w_{ji}. In Section 6 we will analyze atrophy due to disuse learning for these three types of transmission.

Initially,

$$w_{ji}(0) = 1 \tag{15}$$

for $i = 1, \ldots, M$ and $j = 1, \ldots, N$. The learning law (14) implies that a path weight w_{ji} can start to decay when the total top-down signal σ_i to the ith target F_1 node exceeds the node's activity x_i. The rate of decay is proportional to a path's contribution, S_{ji}, to the top-down signal. Note that if the F_1 pattern $\mathbf{x}$ and the F_2 pattern $\mathbf{y}$ are constant during a learning interval, and if $\sigma_i > x_i$ at the start of that interval, then one or more weights w_{ji} must continue to decay until σ_i converges to x_i. As some S_{ji} fall toward 0, the corresponding weights w_{ji} will cease changing. However, because σ_i is the sum of signals S_{ji}, at least one w_{ji} will continue to fall until $\sigma_i \twoheadrightarrow x_i$. In fact,

$$\frac{d}{dt}\left(\sum_{j=1}^{N} w_{ji}\right) = -\sigma_i(\sigma_i - x_i). \tag{16}$$

When F_2 makes a choice, by eqn (2), we will see that:

$$\sigma_i = S_{Ji} = w_{Ji}, \tag{17}$$

while $S_{ji} = 0$ ($j \neq J$), for all three transmission rules. In this case the atrophy due to disuse eqn (14) reduces to:

$$\begin{aligned}\frac{dw_{ji}}{dt} &= -S_{ji}(w_{Ji} - x_i)\\ &= \begin{cases} -w_{Ji}(w_{Ji} - x_i) & \text{if } j = J \\ 0 & \text{if } j \neq J. \end{cases}\end{aligned} \tag{18}$$

Comparing eqn (18) with eqn (16) illustrates the sense in which the total weighted signal σ_i in a distributed code replaces the weight w_{Ji} in a system where F_2 makes a choice. Note that w_{Ji} approaches x_i at a rate proportional to w_{Ji}. Equation (18) is thereby slightly different from the outstar eqn (1), which reduces to:

$$\frac{dw_{ji}}{dt} = \begin{cases} -(w_{Ji} - x_i) & \text{if } j = J \\ 0 & \text{if } j \neq J \end{cases} \tag{19}$$

when F_2 makes a choice. Because $w_{Ji} = \sigma_i \geq x_i$, $x_i = 0$ if $w_{Ji} = 0$. Thus, eqns (18) and (19) both imply that $\mathbf{w}_J \twoheadrightarrow \mathbf{x}$ while other $\mathbf{w}_j$ remain constant, as long as the Jth F_2 node remains active. With fast learning, the two laws are equivalent. Therefore, neither computational nor experimental analysis of such a system, with choice at F_2 and fast learning, can differentiate outstar learning from atrophy due to disuse. The three synaptic transmission rules are similarly indistinguishable. However, when F_2 activity $\mathbf{y}$ is distributed, qualitative properties of learned patterns depend critically on both the learning law and the signal transmission rule, as follows.

5. SYNAPTIC TRANSMISSION FUNCTIONS

We will analyze computational properties of three rules for synaptic transmission. The F_2 path signal vector $\mathbf{y} = (y_1, \ldots y_j, \ldots y_N)$ is assumed to be normalized:

$$\sum_{j=1}^{N} y_j = 1, \tag{20}$$

but is otherwise arbitrary. Given a signal y_j from the jth F_2 node to the ith F_1 node, via a path with an adaptive weight w_{ji}, the net signal S_{ji} received by the ith F_1 node is assumed to be a function of y_j and w_{ji}:

$$S_{ji} = f(y_j, w_{ji}). \tag{21}$$

Each of the three rules that will now be considered corresponds to a physical theory of synaptic signal transmission in neural pathways. The present analysis uses computational considerations alone to select one of these three rules over the others in a neural system for spatial pattern learning.

The first synaptic transmission rule postulates that the $F_2 \twoheadrightarrow F_1$ signal is jointly proportional to the path signal y_j and the weight w_{ji}:

Product rule

$$S_{ji} = y_j w_{ji}. \tag{22}$$

Synaptic transmission by the product rule is an implied hypothesis of a large majority of neural network models. The rule implies that σ_i, the sum of all transmitted signals to the ith F_1 node, equals the dot product between the $F_2 \twoheadrightarrow F_1$ path vector $(y_1, \ldots y_j, \ldots y_N)$ and the converging weight vector $(w_{1i}, \ldots w_{ji}, \ldots, w_{Ni})$. That is, the total signal from F_2 to the ith F_1 node is a linear combination of the path signals y_j:

$$\sigma_i = \sum_{j=1}^{N} y_j w_{ji}, \tag{23}$$

with the coefficients w_{ji} fixed (McCulloch & Pitts, 1943) or determined by some learning law. The total transmitted signal σ_i thereby computes the correlation between the $F_2 \twoheadrightarrow F_1$ path vector and the converging weight vector. Rosenblatt (1962) considered synaptic transmission rules in the general form eqn (21) when defining the perceptron. However, the product rule (22) and its linear matched filter (23) have since come into almost universal use.

A different synaptic transmission rule assumes that the path signal y_j is itself transmitted directly to the ith F_1 node until an upper bound on the path's capacity is reached. With this upper bound equal to the path weight w_{ji}, the net signal obeys the:

Capacity rule

$$S_{ji} = y_j \wedge w_{ji} \equiv \min(y_j, w_{ji}). \tag{24}$$

A capacity rule is suggested by the computational requirements of neural network realizations of fuzzy set theory, as in fuzzy ART (Carpenter et al., 1991b; Carpenter & Grossberg, 1993). Figure 2 illustrates how the product rule compares to the capacity rule. For each, the signal S_{ji} grows linearly when y_j is small.

However, a product rule signal increases with y_j for all $y_j \in [0, 1]$, and a capacity rule signal ceases to grow when y_j reaches the upper bound w_{ji}.

The geometry of the graph in Figure 2 suggests consideration of a third signal function, to complete a transmission rule parallelogram. The third signal describes a:

Threshold rule

$$S_{ji} = [y_j - (1 - w_{ji})]^+. \tag{25}$$

It is awkward to try to interpret eqn (25) in terms of the weight w_{ji}. However, a natural interpretation can be made if the unit of long-term memory is taken to be a signal threshold τ_{ji} rather than the path weight w_{ji}. Namely, by setting:

$$\tau_{ji} \equiv 1 - w_{ji}, \tag{26}$$

the threshold rule (25) becomes:

$$S_{ji} = [y_j - \tau_{ji}]^+ \tag{27}$$

In eqn (27), the transmitted signal from the jth F_2 node to the ith F_1 node is the amount by which the path signal y_j exceeds an adaptive synaptic threshold τ_{ji}.

Note that the three rules (22), (24), and (25) are identical if F_2 activity is binary, because for each rule:

$$S_{ji} = \begin{cases} w_{ji} & \text{if} \quad y_j = 1 \\ 0 & \quad y_j = 0. \end{cases} \tag{28}$$

In particular, the three synaptic transmission rules are computationally indistinguishable if F_2 makes a choice, by eqn (2). However, when a normalized F_2 code is distributed, an adaptive system that uses either the product rule or the capacity rule can suffer catastrophic forgetting. The threshold rule solves this problem.

6. PATH WEIGHTS VERSUS SIGNAL THRESHOLDS AS THE UNIT OF LONG-TERM MEMORY

We will analyze atrophy due to disuse learning laws when S_{ji} is described by one of the three synaptic transmission rules, listed in Table 1.

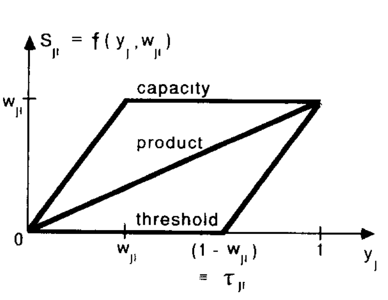

FIGURE 2. A synaptic transmission parallelogram. S_{ji} is the transmitted signal from the jth F_2 node to the ith F_1 node. (a) By the product rule, $S_{ji} = y_j w_{ji}$. (b) By the capacity rule, $S_{ji} = y_j \wedge w_{ji}$. (c) By the threshold rule, $S_{ji} = [y_j - (1 - w_{ji})]^+ = [y_j - \tau_{ji}]^+$. The three rules agree when y is a binary code.

TABLE 1
Synaptic Transmission Functions

(a) Product rule:	$S_{ji} = y_j w_{ji}$	(22)
(b) Capacity rule	$S_{ji} = y_j \wedge w_{ji}$	(24)
(c) Threshold rule.	$S_{ji} = [y_j - (1 - w_{ji})]^+$	(25)

Note that eqn (14) could also be used for spatial pattern learning in a system where x_i may be greater than σ_i. Then, the top-down signal vector σ would still track the F_1 spatial pattern vector $\mathbf{x}$. However, the top-down prime hypothesis (4) implies that weights can only decrease, and hence are guaranteed to converge to some limit in the interval [0, 1] for arbitrary learning and input regimes.

Consider an atrophy due to disuse system (14) in its initial state, when no learning has yet taken place. Then, all $w_{ji} = 1$. Thus, for each of the three synaptic transmission rules (Table 1):

$$S_{ji}(0) = y_j(0) \tag{29}$$

Therefore, because the F_2 activity vector $\mathbf{y}$ is normalized, by eqn (20),

$$\sigma_i(0) = \sum_{j=1}^{N} S_{ji}(0) = 1. \tag{30}$$

Suppose that $x_i = \sigma_i \wedge I_i$, as in eqn (9). Then

$$x_i(0) = I_i \in [0, 1], \tag{31}$$

by eqn (30). Moreover, eqns (14) and (30) imply that x_i will remain equal to I_i for as long as $\mathbf{I}$ remains constant. During that time, as some or all weights w_{ji} decrease, the top-down input σ_i will decay toward the bottom-up input I_i, no matter which transmission rule is selected. For each rule,

$$\frac{d}{dt} w_{ji} = -S_{ji}(\sigma_i - I_i) \tag{32}$$

When F_2 makes a choice, as in eqn (2), $\sigma_i = w_{Ji}$, which converges toward I_i, by eqn (32). All other weights w_{ji} ($j \neq J$) remain constant. Competition at F_2 hereby limits the maximum total weight change at each F_1 node. In fact, when F_2 makes a choice,

$$\begin{aligned} \Delta\left(\sum_{j=1}^{N} w_{ji}\right) &\equiv \sum_{j=1}^{N} [w_{ji}(0) - w_{ji}(\infty)] \\ &= [w_{Ji}(0) - w_{Ji}(\infty)] = (1 - I_i) \end{aligned} \tag{33}$$

for all three signal transmission rules.

An F_2 code is maximally compressed when the system makes a choice. Consider now the opposite extreme, when an F_2 code is maximally distributed. That is, let:

$$y_j = \frac{1}{N} \tag{34}$$

for $j = 1, \ldots, N$. All weights $w_{1i}, \ldots, w_{Ni}$ obey eqn (32) and all are initially equal, by eqn (15). Therefore the weights w_{ji} ($j = 1, \ldots, N$) to a given F_1 node will remain equal to one another during learning, for any transmission function S_{ji}. However, these individual weight changes under the three transmission rules show important qualitative differences, despite the fact that the total $F_2 \to F_1$ signal vector σ correctly learns the F_1 activity vector $\mathbf{x} = \mathbf{I}$ for all three. In particular, the nature of the pattern encoded by a given weight vector and the size of the total weight change at each F_1 node clearly distinguish the three rules, as follows.

With the product rule (22),

$$S_{ji} = \frac{1}{N} w_{ji}. \tag{35}$$

Therefore:

$$\sigma_i = \sum_{j=1}^{N} \frac{1}{N} w_{ji} = \frac{1}{N} \sum_{j=1}^{N} w_{ji} \tag{36}$$

and

$$\frac{d}{dt} w_{ji} = -\frac{1}{N} w_{ji}\left(\frac{1}{N} \sum_{k=1}^{N} w_{ki} - I_i\right). \tag{37}$$

Because all weights w_{ji} to the ith F_1 node ($j = 1, \ldots, N$) remain equal during learning,

$$w_{ji} \to I_i. \tag{38}$$

Thus, the maximum total weight change at an F_1 node i is

$$\Delta\left(\sum_{j=1}^{N} w_{ji}\right) = N(1 - I_i), \tag{39}$$

which could be anywhere from 0 (when $I_i = 1$) to N (when $I_i = 0$).

With the capacity rule (24),

$$S_{ij} = \frac{1}{N} \wedge w_{ji} = \begin{cases} \frac{1}{N} & \text{if } \frac{1}{N} \le w_{ji} \le 1 \\ w_{ji} & \text{if } 0 \le w_{ji} \le \frac{1}{N}. \end{cases} \tag{40}$$

Therefore:

$$\sigma_i = \begin{cases} 1 & \text{if } \frac{1}{N} \le w_{ji} \le 1 \quad \text{for all } j \\ \sum_{j=1}^{N} w_{ji} & \text{if } 0 \le w_{ji} \le \frac{1}{N} \quad \text{for all } j. \end{cases} \tag{41}$$

Equation (41) accounts for all cases because $w_{1i} = \ldots = w_{Ni}$ during learning. Weights adapt according to:

$$\frac{d}{dt} w_{ji} = \begin{cases} -\frac{1}{N}(1 - I_i) & \text{if } \frac{1}{N} \le w_{ji} \le 1 \\ -w_{ji}\left(\sum_{k=1}^{N} w_{ki} - I_i\right) & \text{if } 0 \le w_{ji} \le \frac{1}{N}. \end{cases} \tag{42}$$

By eqn (42), unless $I_i = 1$, all weights w_{ji} shrink until they enter the interval $[0, 1/N]$. Thus:

$$w_{ji} \to \begin{cases} \frac{I_i}{N} & \text{if } 0 \le I_i < 1 \\ 1 & \text{if } I_i = 1 \end{cases} \tag{43}$$

for each $j = 1, \ldots, N$. The maximum total weight change at the ith F_1 node is:

$$\Delta\left(\sum_{j=1}^{N} w_{ji}\right) = \begin{cases} (N - I_i) & \text{if } 0 \le I_i < 1 \\ 0 & \text{if } I_i = 1 \end{cases} \tag{44}$$

which lies between $(N - 1)$ and N, unless $I_i = 1$.

With the threshold rule (25),

$$S_{ji} = \begin{cases} \left[\frac{1}{N} - (1 - w_{ji})\right] & \text{if } \left(1 - \frac{1}{N}\right) \le w_{ji} \le 1 \\ 0 & \text{if } 0 \le w_{ji} \le \left(1 - \frac{1}{N}\right). \end{cases} \tag{45}$$

By eqns (14) and (45), weight w_{ji} would cease to change if it fell to $(1 - 1/N)$. Thus, because all $w_{ji}(0) = 1$,

$$\sigma_i = 1 - \sum_{j=1}^{N} (1 - w_{ji}). \tag{46}$$

During learning,

$$\frac{d}{dt} w_{ji} = -\left[\frac{1}{N} - (1 - w_{ji})\right] \times \left[1 - \sum_{k=1}^{N} (1 - w_{ki}) - I_i\right], \tag{47}$$

so:

$$\sum_{j=1}^{N} w_{ji} \to N - (1 - I_i). \tag{48}$$

Therefore, because weights to the ith node remain equal as they decay:

$$w_{ji} \to 1 - \left(\frac{1 - I_i}{N}\right). \tag{49}$$

In other words, the threshold $\tau_{ji} \equiv 1 - w_{ji}$ rises from 0 until:

$$\tau_{ji} \to \left(\frac{1 - I_i}{N}\right). \tag{50}$$

Thus, $\tau_{ji} \in [0, 1/N]$ after learning. The total weight change at the ith node is:

$$\Delta\left(\sum_{j=1}^{N} w_{ji}\right) = (1 - I_i). \tag{51}$$

Like the weights, the maximum total threshold change at the ith node equals $(1 - I_i)$.

Compare now the different asymptotic weights for the three synaptic transmission rules learned under the maximally distributed F_2 code (34). Although for all three rules the total top-down signal σ_i converges to

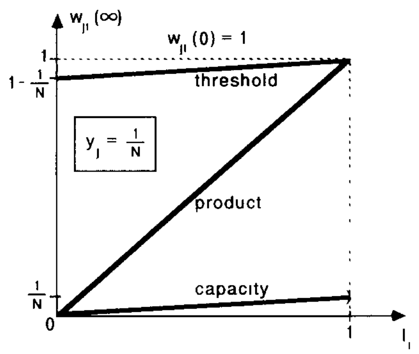

FIGURE 3. Asymptotic weight values for a fully distributed code, where $y_j = 1/N$. As a function of I_i, the dynamic range of $w_{ji}(\infty)$ depends critically upon the choice of synaptic transmission rule: (a) product rule, (b) capacity rule, or (c) threshold rule. During learning, weights decrease, from an initial value of $w_{ji}(0) = 1$, except when $I_i = 1$.

the bottom-up signal I_i at each F_1 node i, the total weight change varies dramatically (Figure 3). Recall that when F_2 makes a choice the maximum total weight change at a given node equals $(1 - I_i) \in [0, 1]$ for all three rules. With distributed F_2 activity and a product rule, all weights w_{ji} converge to I_i and the maximum total weight change is $N(1 - I_i) \in [0, N]$. The full range of all weight values is thus spanned upon presentation of the very first input. In particular, all weights w_{ji} ($j = 1, \ldots, N$) to the ith F_1 node decay to 0 if $I_i = 0$. Because weight values can only decrease during learning, these weights would remain at 0 for all time. Moreover, the top-down prime hypothesis (4) implies that F_1 activity x_i would always be zero for any future input **I** and any F_2 code **y**. Thus, the fact that a single component was zero on just one input interval would render that component useless for all future input presentations, unable to be registered in LTM or even in STM. Similarly each I_i value of the first input would set an upper bound on all future x_i values, because

$$x_i \le \sigma_i = \sum_{j=1}^{N} y_j w_{ji} \le I_i \sum_{j=1}^{N} y_j = I_i \tag{52}$$

for any F_2 code **y**. If a sequence of inputs $\mathbf{I}^{(1)}, \mathbf{I}^{(2)}, \ldots$ were to activate the distributed code (34), each weight w_{ji} would converge toward the minimum of $I_i^{(1)}, I_i^{(2)}, \ldots$. Within a few input presentations, all weights w_{ji} would, in all likelihood, decay toward zero. Similar problems occur for other distributed codes **y**. In this sense, the product rule leads to catastrophic forgetting.

The situation with the capacity rule is even worse (Figure 3). When the F_2 code is fully distributed, all weights w_{ji} decay to $I_i/N \in [0, 1/N]$, unless $I_i = 1$; and the maximum total weight change at the ith node is $N(1 - I_i)$. Thus, unless **I** is a binary vector, the full dynamic range of weight values is nearly exhausted upon the first input presentation.

It is the adaptive threshold rule alone that limits the total weight change to $(1 - I_i) \in [0, 1]$ for maximally distributed as well as maximally compressed codes **y**. In fact, if **y** is *any* F_2 code that becomes active when all w_{ji} are initially equal to 1, then:

$$w_{ji} \rightarrow 1 - y_j(1 - I_i), \tag{53}$$

as in eqn (49). Equivalently:

$$\tau_{ji} \rightarrow y_j(1 - I_i), \tag{54}$$

by eqn (26). Thus, the total weight/threshold change at each F_1 node i is bounded by $(1 - I_i)$ for any code, provided only that **y** is normalized. An F_2 code **y** would typically be highly distributed, with all y_j close to $1/N$, when a system has no strong evidence to choose one category j over another. In this case, the change of each threshold τ_{ji} is automatically limited to the narrow interval $[0, y_j]$, reserving most of the dynamic range for subsequent encoding. Only when evidence strongly supports selection of the F_2 category node J over all others, with y_J therefore close to 1, would weights be allowed to vary across most of their dynamic range. In particular, it is only when y_J is close to 1 that a weight w_{Ji} is able to drop, irreversibly, toward 0, if I_i is small. Even with fast learning, other weights w_{ji} to the ith node then remain large, even if $y_j > 0$. This is because, by eqns (14) and (25), weight changes cease altogether when:

$$y_j \le 1 - w_{ji} \equiv \tau_{ji} \tag{55}$$

The adaptive threshold τ_{ji} thereby replaces strong F_2 competition as the guardian, or stabilizer, of previously learned codes.

7. DISTRIBUTED OUTSTAR LEARNING

The analysis of distributed spatial pattern learning leads to the selection of a synaptic transmission rule with an adaptive threshold. In terms of the threshold τ_{ji} in the path from the jth F_2 node to the ith F_1 node, a stable learning law for distributed codes is defined as the:

Distributed outstar

$$\frac{d\tau_{ji}}{dt} = S_{ji}(\sigma_i - x_i), \tag{56}$$

where S_{ji} is the thresholded path signal $[y_j - \tau_{ji}]^+$ transmitted from the jth F_2 node to the ith F_1 node and σ_i is the sum:

$$\sigma_i \equiv \sum_{j=1}^{N} S_{ji} = \sum_{j=1}^{N} [y_j - \tau_{ji}]^+. \tag{57}$$

Initially,

$$\tau_{ji}(0) = 0. \tag{58}$$

In a system such as ART 1 or fuzzy ART, where F_1 dynamics are defined so that the total top-down signal σ_i is always greater than or equal to x_i, the distributed

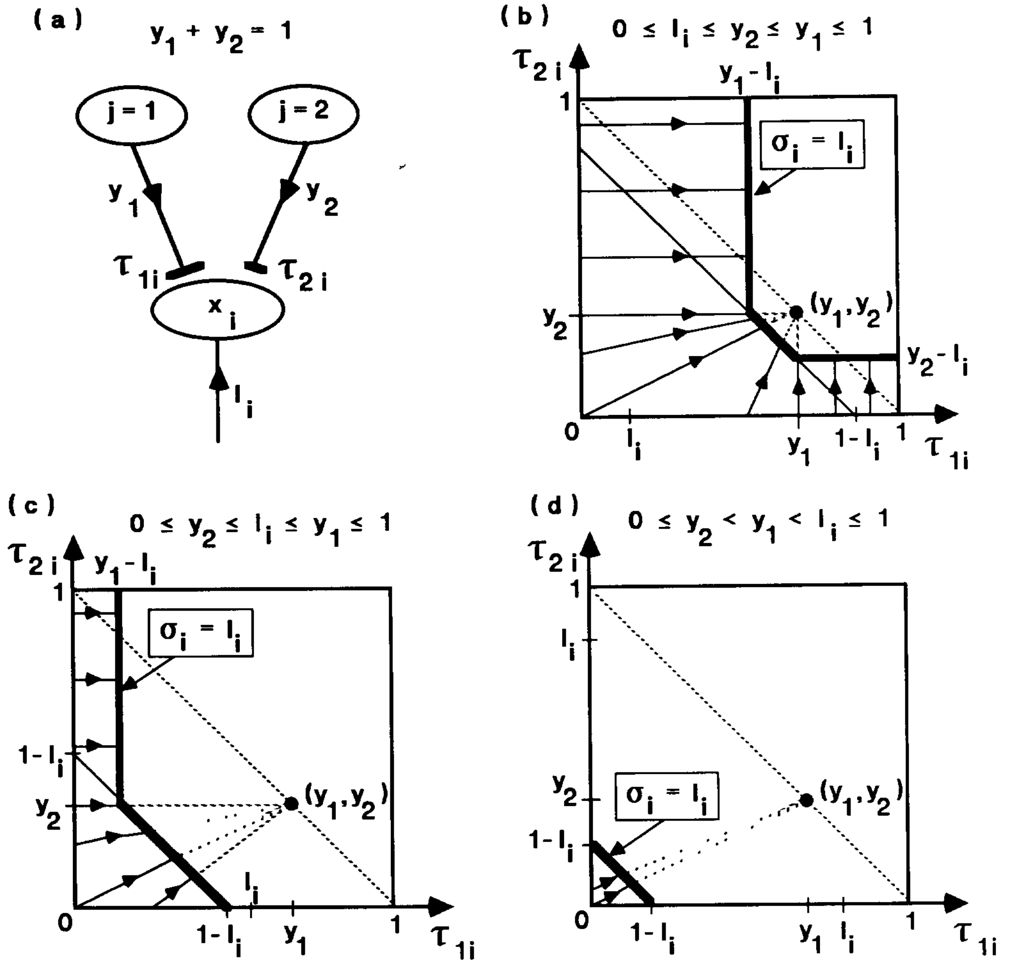

FIGURE 4. (a) A distributed outstar whose coding field F_2 has just two nodes ($N = 2$). For each code y, $y_1 + y_2 = 1$; and $x_i = I_i \wedge \sigma_i$. When thresholds start out small enough, τ_{1i} and/or τ_{2i} increase toward $\{(\tau_{1i}, \tau_{2i}): \sigma_i = I_i\}$. Threshold changes are greatest for small I_i (b). When $I_i > y_j$, the jth node cannot dominate learning (c). When I_i is large, only small thresholds can change at all (d).

outstar allows thresholds τ_{ji} to grow but never shrink. The principle of atrophy due to disuse implies that a threshold τ_{ji} is unable to change at all unless (i) the path signal y_j exceeds the previously learned value of τ_{ji}; and (ii) the total top-down signal σ_i to the ith node exceeds that node's activity x_i. In particular, if τ_{ji} grows large when the node j represents part of a compressed F_2 code, then τ_{ji} cannot be changed at all when node j is later part of a more distributed code, because threshold changes are disabled if $y_j \le \tau_{ji}$.

8. DISTRIBUTED OUTSTAR DYNAMICS

The dynamics of distributed outstar learning will now be illustrated by means of a low-dimensional example. Consider a coding network with just two F_2 nodes (Figure 4a). Two top-down paths, with thresholds τ_{1i} and τ_{2i}, converge upon each F_1 node. Assume that $x_i = I_i \wedge \sigma_i$, as in eqn (9), and fix an F_2 code $\mathbf{y} = (y_1, y_2)$, with:

$$0 \le y_2 \le y_1 \le 1. \tag{59}$$

By the F_2 normalization hypothesis (20), $y_1 + y_2 = 1$. By eqns (27) and (56), for $j = 1, 2$:

$$\frac{d}{dt}\tau_{ji} = [y_j - \tau_{ji}]^+[\sigma_i - I_i]^+, \tag{60}$$

where, by eqn (5),

$$\sigma_i = [y_1 - \tau_{1i}]^+ + [y_2 - \tau_{2i}]^+. \tag{61}$$

Figure 4b–d shows the 2-D phase plane dynamics of the threshold vector (τ_{1i}, τ_{2i}) for a fixed input I_i. In each plot, trajectories that begin in the set of points where $\sigma_i > I_i$ approach the set where $\sigma_i = I_i$. As t increases, the point $(\tau_{1i}(t), \tau_{2i}(t))$ moves along a straight line from small $(\tau_{1i}(0), \tau_{2i}(0))$ toward (y_1, y_2), slowing down asymptotically as:

$$\sigma_i = [y_1 - \tau_{1i}(t)]^+ + [y_2 - \tau_{2i}(t)]^+ \rightarrow I_i. \tag{62}$$

Only if $I_i = 0$ does (τ_{1i}, τ_{2i}) approach (y_1, y_2). Larger thresholds τ_{ji}, which make $\sigma_i \le I_i$, are unchanged during learning. Small I_i allow the greatest threshold changes (Figure 4b). If $I_i = 0$,

$$\tau_{ji} \rightarrow y_j \tag{63}$$

as σ_i decreases to 0. Both thresholds grow if both are initially small. However, if one threshold is so large as to prevent $F_2 \rightarrow F_1$ signal transmission in the corresponding path, the other F_2 node takes over the code. For example, if $\tau_{2i}(0) \geq y_2$ there will be no signal from the F_2 node $j = 2$ to the ith F_1 node, and hence no threshold change in that path. If, then, $\tau_{1i}(0) < y_1 - I_i$, τ_{1i} will increase until:

$$\sigma_i = y_1 - \tau_{1i} \rightarrow x_i = I_i. \tag{64}$$

Larger I_i values permit threshold changes only for smaller initial threshold values. In Figure 4c, τ_{2i} can change only if τ_{1i} changes as well, when both are initially small. In contrast, because y_1 is greater than I_i, τ_{1i} may increase, by itself, toward $(y_1 - I_i)$. Finally, for I_i close to 1 (Figure 4d) adaptive changes can occur only if both τ_{1i} and τ_{2i} are initially small, as they are before any learning has taken place.

REFERENCES

Carpenter, G A., & Grossberg, S (1987a) A massively parallel architecture for a self-organizing neural pattern recognition machine *Computer Vision, Graphics, and Image Processing,* **37,** 54–115 [Reprinted in G A Carpenter & S Grossberg (Eds) (1991) *Pattern recognition by self-organizing neural networks* (pp. 316–382) Cambridge, MA MIT Press]

Carpenter, G A , & Grossberg, S (1987b) ART 2 Stable self-organization of pattern recognition codes for analog input patterns *Applied Optics,* **26,** 4919–4930. [Reprinted in G A Carpenter & S. Grossberg (Eds) (1991) *Pattern recognition by self-organizing neural networks* (pp 398–423) Cambridge, MA MIT Press]

Carpenter, G A , & Grossberg, S (1990). ART 3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures *Neural Networks,* **3,** 129–152 [Reprinted in G A Carpenter & S Grossberg (Eds) (1991) *Pattern recognition by self-organizing neural networks* (pp 451–499) Cambridge, MA MIT Press]

Carpenter, G. A , & Grossberg, S (1993) Fuzzy ARTMAP. A synthesis of neural networks and fuzzy logic for supervised categorization and nonstationary prediction. In R R Yager & L A Zadeh (Eds), *Fuzzy sets, neural networks, and soft computing* New York Van Nostrand Reinhold

Carpenter, G A , Grossberg, S , & Reynolds, J. H (1991) ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network *Neural Networks,* **4,** 565–588 [Reprinted in G A. Carpenter & S. Grossberg (Eds.) (1991) *Pattern recognition by self-organizing neural networks* (pp 503–546) Cambridge, MA MIT Press]

Carpenter, G A., Grossberg, S , & Rosen, D B. (1991a) Fuzzy ART. Fast stable learning and categorization of analog patterns by an adaptive resonance system *Neural Networks,* **4,** 759–771

Carpenter, G. A , Grossberg, S., & Rosen, D. B. (1991b) *A neural network realization of fuzzy ART* (Tech. Rep CAS/CNS-TR-91-021) Boston University, Boston, MA

Estes, W K (1955) Statistical theory of spontaneous recovery and regression *Psychological Review,* **62,** 145–154

Grossberg, S (1968a). Some nonlinear networks capable of learning a spatial pattern of arbitrary complexity *Proceedings of the National Academy of Sciences USA,* **59,** 368–372

Grossberg, S (1968b) A prediction theory for some nonlinear functional-differential equations, I. Learning of lists *Journal of Mathematical Analysis and Applications,* **21,** 643–694

Grossberg, S (1969). Some networks that can learn, remember, and reproduce any number of complicated space-time patterns, I. *Journal of Mathematics and Mechanics,* **19,** 53–91

Grossberg, S (1972) Pattern learning by functional-differential neural networks with arbitrary path weights In K Schmitt (Ed), *Delay and functional-differential equations and their applications* (pp 121–160) New York Academic Press [Reprinted in S Grossberg (Ed) (1982). *Studies of mind and brain* (pp. 159–193) Dordrecht, Holland· D Reidel Publishing Co]

Grossberg, S (1973) Contour enhancement, short term memory, and constancies in reverberating neural networks *Studies in Applied Mathematics,* **LII,** 217–257 [Reprinted in S Grossberg (Ed) (1982) *Studies of mind and brain* (pp 334–378) Dordrecht, Holland D. Reidel Publishing Co]

Grossberg, S (1974). Classical and instrumental learning by neural networks In R Rosen & F Snell (Eds.), *Progress in theoretical biology* (Vol. 3, pp 51–141) New York Academic Press. [Reprinted in S Grossberg (Ed) (1982) *Studies in mind and brain* (pp 68–156) Dordrecht, Holland. D Reidel Publishing Co]

Grossberg, S (1976) Adaptive pattern classification and universal recoding, II Feedback, expectation, olfaction, illusions *Biological Cybernetics,* **23,** 187–202 [Reprinted in G A Carpenter & S Grossberg (Eds) (1991) *Pattern recognition by self-organizing neural networks* (pp 283–315) Cambridge, MA MIT Press]

McCulloch, W. S , & Pitts, W. (1943) A logical calculus of the ideas immanent in nervous activity *Bulletin of Mathematical Biophysics,* **5,** 115–133 [Reprinted in J A Anderson & E. Rosenfeld (Eds) (1988) *Neurocomputing Foundations of research* (pp. 18–27) Cambridge, MA MIT Press]

Rosenblatt, F (1958) The perceptron A probabilistic model for information storage and organization in the brain *Psychological Review,* **65,** 386–408. [Reprinted in J A Anderson & E Rosenfeld (Eds) (1988) *Neurocomputing Foundations of research* (pp 92–114) Cambridge, MA MIT Press]

Rosenblatt, F (1962) *Principles of neurodynamics* Washington, DC Spartan Books

Widrow, B , & Hoff, M E (1960) Adaptive switching circuits *1960 IRE WESCON Convention Record* (pp 96–104) New York· IRE, [Reprinted in J A Anderson & E Rosenfeld (Eds.) (1988). *Neurocomputing Foundations of research* (pp 126–134). Cambridge, MA MIT Press]