SPATIAL PATTERN LEARNING, CATASTROPHIC FORGETTING, AND OPTIMAL RULES OF SYNAPTIC TRANSMISSION

Gail A. Carpenter†

Center for Adaptive Systems
and
Department of Cognitive and Neural Systems
Boston University
111 Cummington Street
Boston, Massachusetts 02215

In: D.L. Levine and W.R. Elsberry (Editors)
Optimality in Biological and Artificial Networks.
Hillsdale, NJ: Lawrence Erlbaum Asso.

November, 1993

Technical Report CAS/CNS TR-93-058
Boston, MA: Boston University

† This research was supported in part by 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 Robin Locke for her valuable assistance in the preparation of the manuscript.

ABSTRACT

It is a neural network truth universally acknowledged, that the signal transmitted to a target node must be equal to the product of the path signal times a weight. Analysis of catastrophic forgetting by distributed codes leads to the unexpected conclusion that this universal synaptic transmission rule may not be optimal in certain neural networks. The distributed outstar, a network designed to support stable codes with fast or slow learning, generalizes the outstar network for spatial pattern learning. 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, where the 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 types of synaptic transmission, 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 optimal unit of long-term memory in such a system is a subtractive threshold, rather than a multiplicative weight.

Key words: Spatial pattern learning, distributed code, outstar, adaptive threshold, rectified bias, atrophy due to disuse, transmission function, neural network

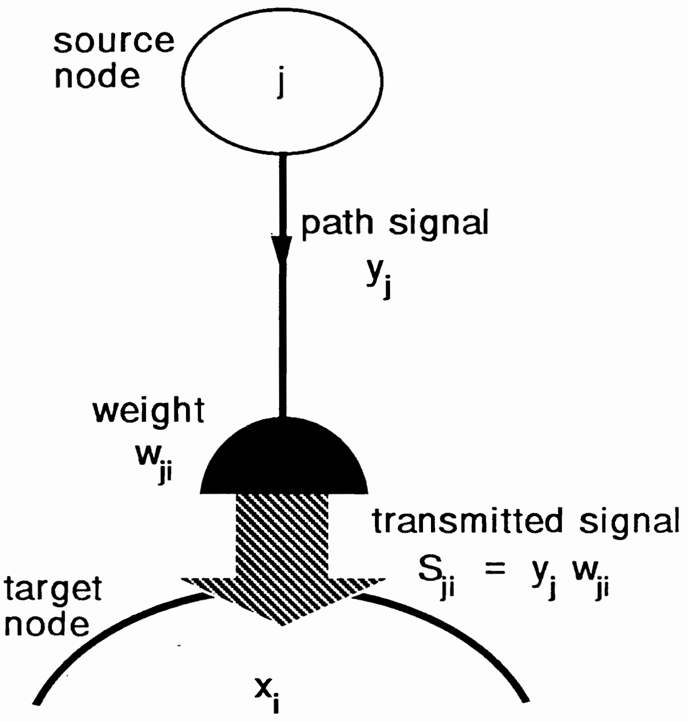

Figure 1. The product rule postulates that the signal transmitted to a target node at a synapse is proportional to a path signal (y_j) times a weight (w_{ji}). This rule is a feature of nearly all neural network models.

1. Optimal rules of synaptic transmission

When neural networks became popular in the 1980s, researchers struggled to define *neural network* with words that include the diverse models in current use. As a step toward this definition, consider the question: What, if anything, do all the neural networks of the past fifty years have in common? The answer to this question is, most likely, nothing. However, the large majority of neural network models, from the McCulloch-Pitts (1943) neuron to the many biological and engineering models at this year's conferences, have at least one thing in common, namely, the rule setting the net signal from a source node to a target node equal to a path signal times a synaptic weight (Figure 1). This *product rule* of synaptic transmission is in such universal use that it is almost always treated as a nameless fact rather than a hypothesis, although neurophysiology so far neither confirms nor refutes this rule. Why, then has this particular process found such widespread use? One answer is its computational power: the product rule sets the sum of weighted signals equal to the dot product of the signal vector and the weight vector. This dot product provides a useful measure of the similarity between the active path signal vector and the learned weight vector. However, utility and universality do not necessarily imply optimality.

This chapter describes a neural network learning problem for which the product rule is not computationally optimal. Solution of the learning problem requires a neural network design to support stable distributed codes. One such design is the *distributed outstar* (Carpenter, 1993, 1994), which solves the distributed code catastrophic forgetting problem when the product rule is replaced by an equally plausible synaptic transmission rule. This *threshold rule* postulates that the unit of long-term memory (LTM) is a subtractive threshold, rather than a multiplicative weight. The computational analysis therefore questions the optimality of a fundamental neural network design hypothesis as it solves a particular learning problem.

2. 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 traces, or adaptive weights, learn stimulus sampling probabilities of border field activity patterns. A sequence of outstars, called an *avalanche*, forms a minimal network for 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 and Grossberg, 1987a, 1987b, 1990; Carpenter, Grossberg, and Rosen, 1991a). The supervised ARTMAP system (Carpenter, Grossberg, and 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 in the neural models that realize the theories, as well as 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 2a) is similar to the original outstar when the source field F_2 contains 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 filter (Figure 3a). 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 here consider how to design a spatial pattern learning network which allows the activity pattern at the coding field F_2 to be arbitrarily distributed (Section 3). 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 4). In particular, if all F_2 nodes were active during learning, all $F_2 \rightarrow F_1$ 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 5). By this principle, a weight in an active path atrophies, or decays, 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 $F_2 \rightarrow F_1$ signal for the individual outstar weight. This seems a plausible step toward spatial pattern learning by a coding source field instead

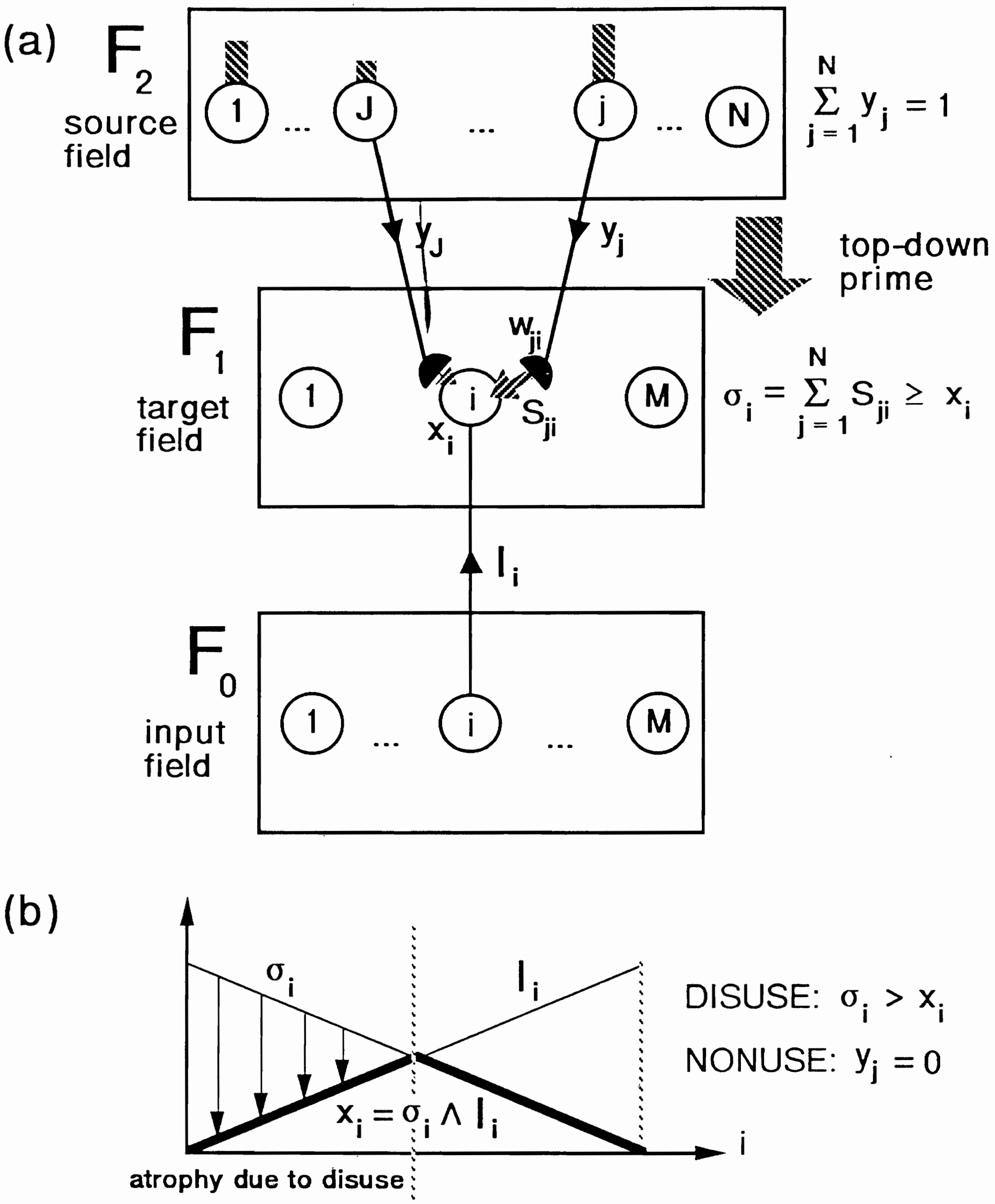

Figure 2. Distributed outstar network for spatial pattern learning. During adaptation a top-down weight w_{ji}, from the j^{th} node of the coding field F_2 to the i^{th} node of the pattern registration field F_1, may decrease or remain constant. An atrophy-due-to-disuse learning law causes the total signal σ_i from F_2 to the i^{th} 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.

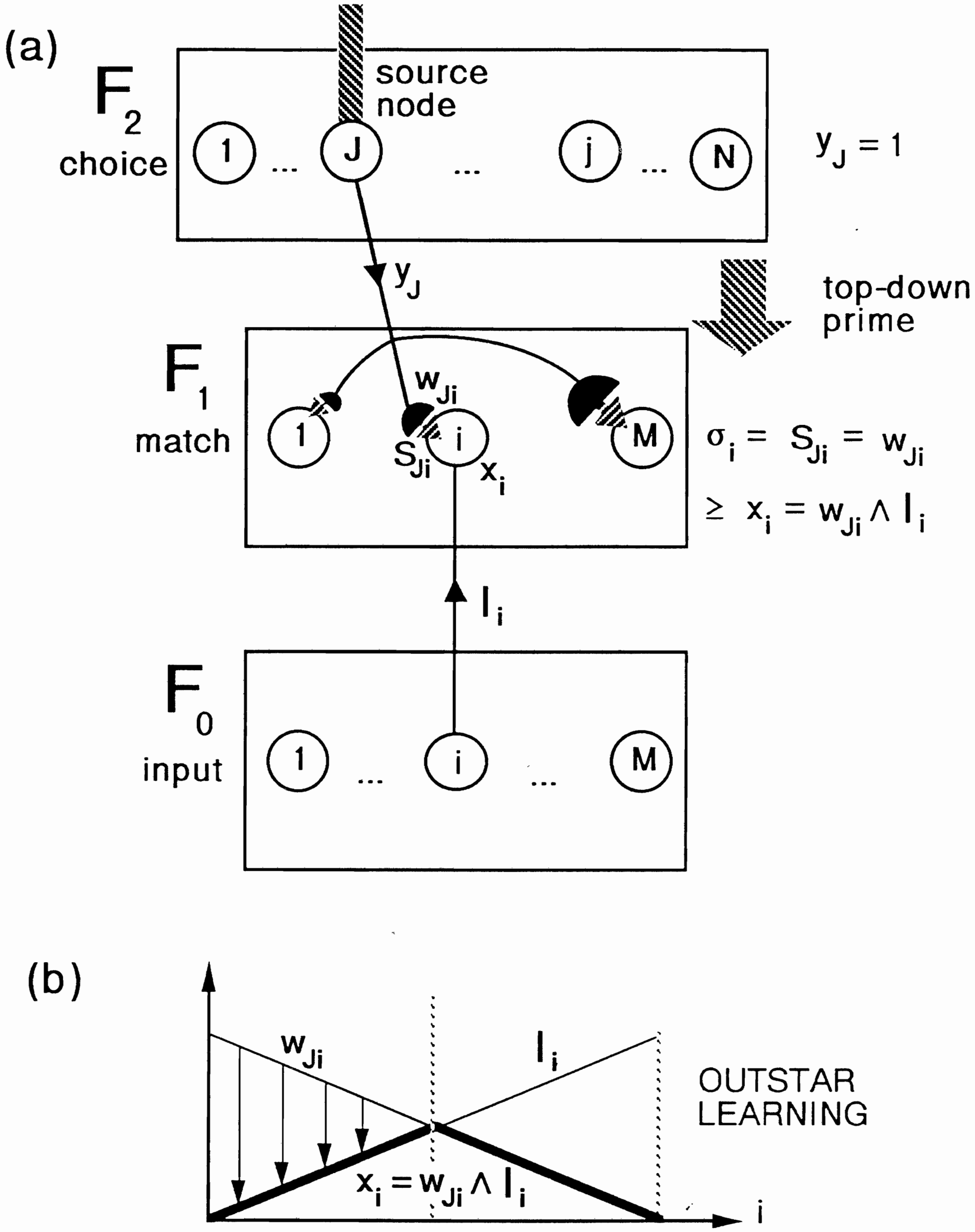

Figure 3. ART 1/fuzzy ART.

of by a single source node. Unfortunately, this development is, by itself, insufficient. 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 j^{th} F_2 node to the i^{th} F_1 node is proportional to the product of the path signal y_j and the path weight w_{ji}. An alternative transmission process, used in a neural network realization of fuzzy ART (Carpenter, Grossberg, and Rosen, 1991b; Carpenter and Grossberg, 1993), obeys a *capacity rule* (Section 6). However, catastrophic forgetting is even more serious a problem for the capacity rule than for the product rule.

Fortunately, another plausible synaptic transmission rule solves the problem (Sections 7–9). This *threshold rule* postulates a transmitted signal equal to the amount by which the $F_2 \rightarrow F_1$ 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 and 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 completes the *distributed outstar learning law*, summarized in Section 10. Section 11 explicitly solves the distributed outstar equations, Section 12 illustrates distributed outstar dynamics with a network that has two nodes in the source field, and Section 11 concludes with a consideration of the physical unit of memory.

3. Spatial pattern learning

The distributed outstar network (Figure 2a) features an adaptive filter from a *coding*, or *source, field* F_2 to a *pattern registration*, or *target, field* F_1. This filter carries 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 is a type of outstar. During outstar learning, weights in the paths emanating from an F_2 node track F_1 activity. That is, when the j^{th} 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 (Figure 3).

While 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 in the top-down adaptive filters of ART 1 (Carpenter and Grossberg, 1987a), ART 2 (Carpenter and Grossberg, 1987b), and fuzzy ART (Carpenter, Grossberg, and Rosen, 1991a). By (1), $w_{ji} \to x_i$ when $y_j > 0$. When $y_j = 0$, w_{ji} remains constant. The term $y_j x_i$ in (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-").

The distributed outstar network does not constitute a stand-alone pattern recognition system. Like the outstar, this module would typically 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 $F_2 \to F_1$ filter plays a crucial role in ART code stabilization. Additional network elements determine which F_2 code will be selected by an input **I** in the first place and implement search and other mechanisms of internal dynamic control (Carpenter and Grossberg, 1987a). This chapter focuses only on design issues pertaining to the top-down adaptive filter.

4. Catastrophic forgetting

The distributed outstar network for spatial pattern learning (Figure 2a) needs 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 i^{th} F_1 node is inactive ($x_i = 0$) due, say, to the fact that there is no input to that node at that moment ($I_i = 0$). With fast learning, an outstar (1) would send all weights w_{ji} ($j = 1, \ldots, N$) to 0. Within an ART system, stability requirements imply that these weights then remain 0 forever. Moreover, no future input I_i to the i^{th} 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 sidestep 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 J^{th} node is described by:

F$_2$ choice

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

In this case, each F_2 node is identified with a class, or category, of inputs **I**. Outstar learning (1) permits a weight w_{ji} to change only if the j^{th} F_2 node is active. When F_2 chooses the node J, all other nodes ($j \neq J$) are inactive. Only the weight w_{Ji} tracks activity at the

i^{th} F_1 node, so:

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

Even if w_{Ji} decays to 0, all other weights to the i^{th} F_1 node remain unchanged when the J^{th} category is selected. These other weights w_{ji} $(j \neq J)$ are thus reserved and can learn their own F_1 patterns when they later become active.

Choice represents an extreme form of short-term memory (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 extremely short. Then, since 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 that is independent of all prior learned differences.

An 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} is an increasing function both of y_j, as in the outstar equation (1), and also of w_{ji} itself. When w_{ji} becomes too small, further change is disallowed. Weights, initially large, can only decrease during learning. Small weights can decrease further only when y_j is close to 1, which occurs when most of the F_2 STM 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, tracks target node activity during learning. A principle of atrophy due to disuse governs weight change, as described in the next section. Within this context, three signal transmission rules are examined (Section 6). An adaptive threshold rule for synaptic transmission is more computationally successful than either of the other two rules.

5. Learning by atrophy due to disuse

The principle of atrophy due to disuse postulates that the strength of an active path decays when the path is disused. Active "dis-use" is distinct from passive "non-use" (Figure 2b), where the strength of an inactive path remains constant, as in outstar learning (1) (Figure 3b). To define disuse, a specific class of target fields F_1 are considered. So far, no assumptions about the F_1 activity vector $\mathbf{x}$ have been made. The main hypothesis on F_1 is 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 to the i^{th} F_1 node, a top-down *priming* input σ_i from F_2 is assumed to be necessary for that node to remain active, once F_2 becomes active. This hypothesis is realized by the inequality:

Top-down prime

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

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

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

(Figure 2a). In particular, when F_2 is active but $\sigma_i = 0$, no activity can be registered at the i^{th} F_1 node, for any bottom-up input $I_i \epsilon [0,1]$.

The top-down prime inequality (4) is closely related to the 2/3 Rule of ART (Carpenter and Grossberg, 1987a), which implies that the i^{th} 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 i^{th} F_1 node to be active, when the J^{th} F_2 node is active, only if $I_i = 1$ and if σ_i exceeds a criterion threshold, where:

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

Fuzzy ART (Carpenter, Grossberg, and Rosen, 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 J^{th} F_2 node is chosen (Figure 3a). The symbol $\wedge$ in (7) denotes the fuzzy intersection (Zadeh, 1965). By (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 (4), the *degree of disuse* D_i of the i^{th} F_1 node is defined to be:

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

When (9) defines F_1 activity,

$$\begin{aligned} D_i &= (\sigma_i - I_i \wedge \sigma_i) \\ &= \begin{cases} \sigma_i - I_i & \text{if } \sigma_i \geq I_i \\ 0 & \text{if } \sigma_i \leq I_i \end{cases} \qquad (11) \\ &= [\sigma_i - I_i]^+, \end{aligned}$$

where $[\ldots]^+$ denotes the rectification operator:

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

where $\vee$ denotes the fuzzy union (Zadeh, 1965). In this case, the degree of disuse at the i^{th} 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 $S_{ji} \geq 0$ and $D_i \geq 0$. With the degree of disuse D_i defined by (10), the learning law (13) becomes:

Atrophy due to disuse

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

(Figure 2b). In Section 6 three synaptic transmission rules will each define S_{ji} as a function of y_j and w_{ji}. In Sections 7 and 8 we will analyze atrophy-due-to-disuse learning and catastrophic forgetting for these three rules.

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 decay when the total top-down signal σ_i to the i^{th} 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. By (14), the sum of all weights converging on the i^{th} node obeys the equation:

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

Thus 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.

When F_2 makes a choice, 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 equation (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 (18) with (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 equation (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 (18) and (19) both imply that $\mathbf{w}_J \rightarrow \mathbf{x}$ while other $\mathbf{w}_j$ remain constant, as long as the J^{th} F_2 node remains active (Figure 3b). With fast learning and F_2 choice the atrophy-due-to-disuse and outstar learning laws are equivalent. In this case, neither computational nor experimental analysis 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.

6. Synaptic transmission functions

We will now define three synaptic transmission rules. 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 j^{th} F_2 node to the i^{th} F_1 node, via a path with an adaptive weight w_{ji}, the net signal S_{ji} received by the i^{th} 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 corresponds to a physical theory of synaptic signal transmission in neural pathways. The present analysis uses computation 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 $F_2 \rightarrow F_1$ signal is jointly proportional to the path signal y_j and the weight w_{ji} :

Product rule

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

(Figure 1). Synaptic transmission by the product rule is an implied hypothesis most neural network models. The rule implies that σ_i, the sum of all transmitted signals to the i^{th} F_1 node, equals the dot product between the $F_2 \rightarrow F_1$ 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 i^{th} 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 and Pitts, 1943) or determined by some learning law. The total transmitted signal σ_i thereby computes the correlation between the $F_2 \rightarrow F_1$ path vector and the converging weight vector. Rosenblatt (1962) considered synaptic transmission rules in the general form (21) when defining the perceptron. However, the product rule (22) and its linear matched filter (23) have since come into almost universal use.

A second synaptic transmission rule assumes that the path signal y_j is itself transmitted directly to the i^{th} 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, Grossberg, and Rosen, 1991b; Carpenter and Grossberg, 1993). Figure 4a 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 $y_j \in [0, 1]$, while a capacity rule signal ceases to grow when y_j reaches the upper bound w_{ji}.

The geometry of the graph in Figure 4a suggests a third signal function, to complete a transmission rule parallelogram. The third signal function describes a:

Threshold rule

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

It is awkward to interpret the transmission rule (25) in terms of the weight w_{ji}. However, a natural interpretation takes the unit of long-term memory 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 (27), the transmitted signal from the j^{th} F_2 node to the i^{th} F_1 node is the amount by which the path signal y_j exceeds an adaptive synaptic threshold τ_{ji}.

The three rules (22), (24), and (25) are identical if F_2 activity is binary, since for each rule:

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

In particular, the three synaptic transmission rules are computationally indistinguishable if F_2 makes a choice, by (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.

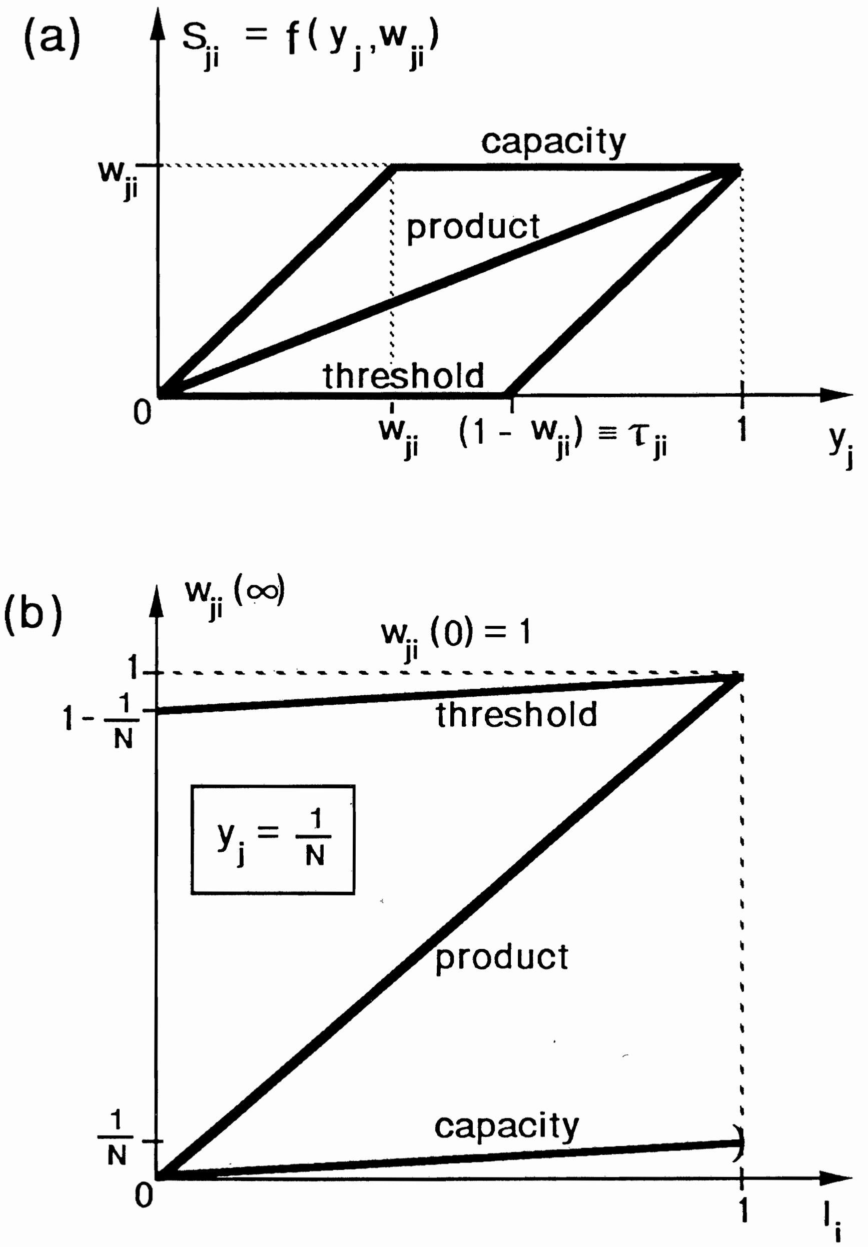

Figure 4. (a) A synaptic transmission parallelogram. S_{ji} is the transmitted signal from the j^{th} F_2 node to the i^{th} F_1 node. By the product rule, $S_{ji} = y_j w_{ji}$. By the capacity rule, $S_{ji} = y_j \wedge w_{ji}$. By the threshold rule, $S_{ji} = [y_j - (1 - w_{ji})]^+ = [y_j - \tau_{ji}]^+$. The three rules agree when **y** is a binary code. (b) Asymptotic weight values for a fully distributed code, where $y_j = \frac{1}{N}$. As a function of I_i, the dynamic range of $w_{ji}(\infty)$ depends critically upon the choice of synaptic transmission rule. During learning, weights decrease, from an initial value of $w_{ji}(0) = 1$, except when $I_i = 1$.

$$\textbf{Product rule}: S_{ji} = y_j w_{ji} \tag{22}$$

$$\textbf{Capacity rule}: S_{ji} = y_j \wedge w_{ji} \tag{24}$$

$$\textbf{Threshold rule}: S_{ji} = [y_j - (1 - w_{ji})]^+ \tag{25}$$

Table 1: Synaptic transmission functions

7. Transmission rule computations

When an F_2 code $\mathbf{y}$ is maximally compressed, the three synaptic transmission rules (Table 1) are computationally identical. Computations in this section demonstrate how the three rules diverge when the F_2 code is maximally distributed. Note that the weight adaptation equation (14) also learns spatial patterns in a system where x_i may sometimes 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.

Initial values: Consider an atrophy-due-to-disuse system (14) in its initial state, when no learning has yet taken place. Then, all $w_{ji} = 1$, so:

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

for each of the three synaptic transmission rules (Table 1). Therefore, since the F_2 activity vector $\mathbf{y}$ is normalized (20),

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

The following computations trace an example in which $x_i = I_i \wedge \sigma_i$, as in (9). Then:

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

by (30). The atrophy-due-to-disuse equation (14) then implies 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 total top-down input σ_i will decay toward the bottom-up input I_i, no matter which transmission rule is selected. For each rule,

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

Choice at F_2: When F_2 makes a choice, as in (2), $\sigma_i = w_{Ji}$, which converges toward I_i, by (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,

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

for all three signal transmission rules.

Distributed code at F_2: 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 equation (32) and all are initially equal, by (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 significant qualitative differences, despite the fact that the total $F_2 \rightarrow F_1$ 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.

Product rule: 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

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

Since all weights w_{ji} to the i^{th} F_1 node remain equal during learning,

$$w_{ji} \rightarrow I_i \tag{38}$$

for $j = 1, \ldots, N$. Thus the maximum total weight change at an F_1 node i is:

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

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

Capacity rule: With the capacity rule (24),

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

Therefore:

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

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

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

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

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

for each $j = 1, \ldots, N$. The maximum total weight change at the i^{th} F_1 node is:

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

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

Threshold rule: With the threshold rule (25),

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

By (14) and (45), weight w_{ji} ceases to change as it falls toward $(1 - \frac{1}{N})$. Thus, since all $w_{ji}(0) = 1$,

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

During learning,

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

so:

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

Therefore, since weights to the i^{th} node remain equal as they decay:

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

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

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

Thus $\tau_{ji} \, \epsilon \, [0, \frac{1}{N}]$ after learning. The total weight change at the i^{th} node is:

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

Like the weights, the sum of all threshold changes at the i^{th} node is less than or equal to $(1 - I_i)$.

8. Transmission rules, catastrophic forgetting, and stable coding

Compare now the different asymptotic weights learned under the maximally distributed F_2 code (34) using the three synaptic transmission rules. For all three rules the total top-down signal σ_i converges to the bottom-up signal I_i at each F_1 node i. However, the total weight changes vary dramatically (Figure 4b), in contrast to the F_2 choice case, where the maximum total weight change at a given node equals $(1 - I_i) \, \epsilon \, [0, 1]$ for all three rules.

Product rule - Catastrophic forgetting: With distributed F_2 activity and a product rule, all weights w_{ji} converge to I_i and the maximum total weight change is $N(1 - I_i) \, \epsilon \, [0, N]$. 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 i^{th} F_1 node decay to 0 if $I_i = 0$. Since weight values can only decrease during learning, these weights would remain equal to 0 for all time. Moreover, the top-down prime hypothesis (4) implies that F_1 activity x_i would then always be zero for any future input $\mathbf{I}$ and any F_2 code $\mathbf{y}$. Thus, the fact that a given 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 $I_i = I_i^{(1)}$ value of the first input would set an upper bound on all future x_i values, since

$$\begin{aligned} x_i \le \sigma_i &= \sum_{j=1}^{N} y_j w_{ji} \\ &\le I_i^{(1)} \sum_{j=1}^{N} y_j = I_i^{(1)} \end{aligned} \tag{52}$$

for any F_2 code $\mathbf{y}$. If a sequence of inputs $\mathbf{I}^{(1)}, \mathbf{I}^{(2)}, \ldots$ were to activate the fully 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. This problem occurs for any distributed code $\mathbf{y}$. In this sense, the product rule leads to catastrophic forgetting.

Capacity rule – Even-more-catastrophic forgetting: The situation with the capacity rule is even worse (Figure 4b). When the F_2 code is fully distributed, all weights w_{Ji} decay to $\frac{I_i}{N} \epsilon [0, \frac{1}{N}]$, unless $I_i = 1$; and the maximum total weight change at the i^{th} node is $N(1 - I_i)$. Thus, unless $\mathbf{I}$ is a binary vector, the entire dynamic range of weight values is nearly exhausted upon the first input presentation.

Threshold rule – Stable coding: It is the adaptive threshold rule alone that limits the total weight change to $(1 - I_i) \epsilon [0, 1]$ for maximally distributed as well as maximally compressed codes $\mathbf{y}$. In fact, if $\mathbf{y}$ 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 (49). Equivalently:

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

by (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 $\mathbf{y}$ is normalized. An F_2 code $\mathbf{y}$ would typically be highly distributed, with all y_j close to $\frac{1}{N}$, 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 i^{th} node then remain large, even if all $y_j > 0$. This is because, by (14) and (25), weight changes cease altogether when:

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

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

9. Confidence-plasticity tradeoff

Figure 5 illustrates why the product rule and the capacity rule cause catastrophic forgetting and how the threshold rule solves this problem. During atrophy-due-to-disuse learning, if the i^{th} F_1 target node is disused $(\sigma_i > x_i)$ then the weight w_{ji} will decay in any path that sends a signal to the i^{th} node $(S_{ji} > 0)$ (Figure 5a). When F_2 makes a choice, each of the three synaptic transmission rules allows weight change in only one path to each target node. However, if y_j is even slightly positive, both the product rule (Figure 5b) and the capacity rule (Figure 5c) allow weights w_{ji} to decay without limit, unless learning rates are very slow. In contrast, the threshold rule (Figure 5d) implies that, even if the J^{th} F_2 node is active, the signal S_{Ji} is still zero if the path threshold is large $(\tau_{Ji} \geq y_J)$; or, equivalently, if the path weight is small $(w_{Ji} \leq 1 - y_J)$. Only the positive signals S_{Ji} sum to σ_i and only these signals can atrophy due to disuse. Threshold τ_{Ji} remains small, and therefore plastic, if y_J is always small when $\sigma_i > x_i$. If y_J is large, τ_{Ji} may increase toward 1. Once this occurs, however, $S_{Ji} = 0$ for all F_2 codes $\mathbf{y}$ except those which compress most activity at the J^{th} node. Thus in a recognition system that allows an F_2 node to become highly active only when it is highly confident of its choice, the threshold rule automatically links confidence to stability. Conversely, when category selection is uncertain, distributed codes retain plasticity.

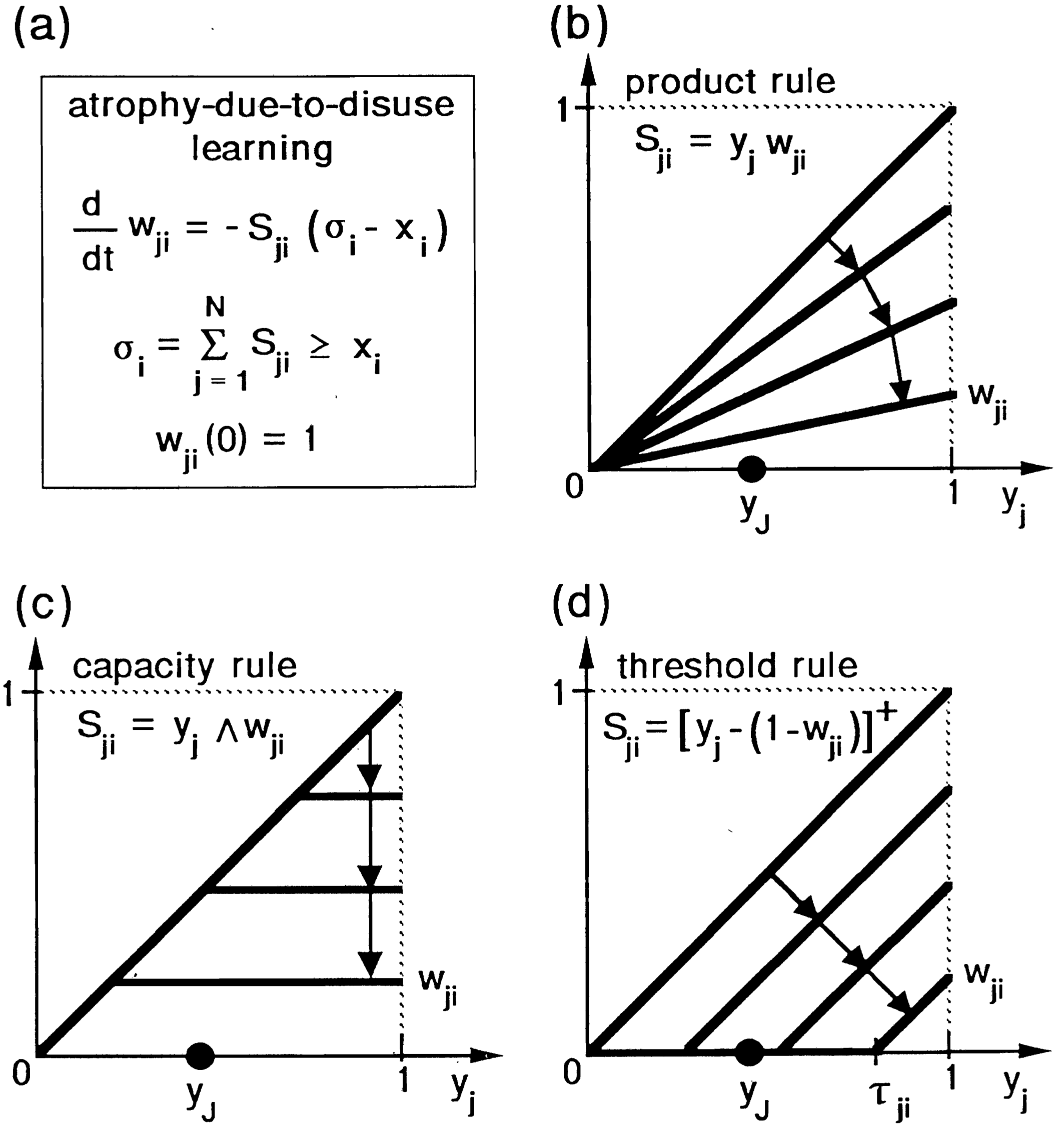
(a)
atrophy-due-to-disuse learning
$\frac{d}{dt} w_{ji} = -S_{ji} (\sigma_i - x_i)$
$\sigma_i = \sum_{j=1}^{N} S_{ji} \geq x_i$
$w_{ji}(0) = 1$
(b)
product rule
$S_{ji} = y_j w_{ji}$
1
0
y_J
1
y_j
w_{ji}
(c)
capacity rule
$S_{ji} = y_j \wedge w_{ji}$
1
0
y_J
1
y_j
w_{ji}
(d)
threshold rule
$S_{ji} = [y_j - (1 - w_{ji})]^+$
1
0
y_J
τ_{ji}
1
y_j
w_{ji}

Figure 5. (a) Atrophy-due-to-disuse learning causes a weight w_{ji} to decay at a rate proportional to (i) the signal from the j^{th} F_2 node to the i^{th} F_1 node and (ii) the degree of disuse, which equals to the difference between the total F_2 — F_1 signal to the i^{th} node and the activity of that node. (b) When the J^{th} F_2 node is active, the product rule implies that that signal S_{Ji} to the i^{th} F_1 node is positive. All weights w_{Ji} therefore decay when $\sigma_i > x_i$, even if those weights are already small. This causes catastrophic forgetting. (c) The capacity rule leads to catastrophic forgetting for the same reason as the product rule. (d) The threshold rule buffers learned codes against catastrophic forgetting by allowing only paths with sufficiently large weights (small thresholds) to contribute to the recognition code and hence to be subject to change during learning.

10. Distributed outstar learning

Computational 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 j^{th} F_2 node to the i^{th} 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 j^{th} F_2 node to the i^{th} 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, the total top-down signal primes F_1. That is, σ_i is always greater than or equal to x_i. The distributed outstar then 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 i^{th} 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, since threshold changes are disabled if $y_j \leq \tau_{ji}$ (Figure 5d).

11. Distributed outstar solution

The form of the distributed outstar system (56) – (58) is so simple that the equations can be solved in closed form. The formulas below give an explicit solution for an arbitrary input sequence with either slow or fast learning. Section 12 illustrates the geometry of this solution.

Assume that an input **I** activates a distributed outstar field F_1 at some time $t = t_0$ and that **I** is held fixed for some ensuing interval. If $\sigma_i \leq x_i$ at $t = t_0$, then τ_{ji} will remain constant during that interval, for all $j = 1, \ldots, N$. Similarly, τ_{ji} will remain constant if $y_j \leq \tau_{ji}$ at $t = t_0$. Consider now a fixed F_1 index i such that $\sigma_i > x_i$ at $t = t_0$. Let:

$$\Phi_i = \{j : y_j(t_0) > \tau_{ji}(t_0)\}. \tag{59}$$

For $j \in \Phi_i$,

$$\frac{d}{dt}\tau_{ji} = (y_j - \tau_{ji})(\sigma_i - x_i), \tag{60}$$

until y_j and x_i change. Geometrically, by (60), the projected vector of τ_{ji} values with $j \in \Phi_i$ follows a straight line toward the corresponding projected vector of y_j values. If all such τ_{ji}

were to approach y_j then σ_i would converge to 0, by (57). Progress halts, however, as the τ_{ji} vector approaches the set of points where $\sigma_i = x_i$, by (60).

Explicitly, for $t \geq t_0$, while y_j and x_i are constant:

$$\tau_{ji}(t) = \tau_{ji}(t_0) + \alpha(t)\frac{[\sigma_i(t_0) - x_i]^+}{\sigma_i(t_0)}[y_j - \tau_{Ji}(t_0)]^+, \tag{61}$$

where $\alpha(t)$ is an exponential that goes from 0 to 1 as t goes from t_0 to ∞.

By (61), $\tau_{ji}(t)$ remains constant if $\sigma_i(t_0) \leq x_i$ or if $y_j \leq \tau_{ji}(t_0)$. If $\sigma_i(t_0) > x_i$ and if $j \in \Phi_i, \tau_{ji}(t)$ moves from $\tau_{ji}(t_0)$ toward:

$$\tau_{ji}(\infty) = \tau_{ji}(t_0) + \frac{(\sigma_i(t_0) - x_i)}{\sigma_i(t_0)}(y_j - \tau_{ji}(t_0)) \tag{62}$$

as t goes from t_0 to ∞. In particular:

$$\begin{aligned}
\sigma_i(\infty) &= \sum_{j\in\Phi_i}(y_j - \tau_j(\infty)) \\
&= \sum_{j\in\Phi_i}(y_j - \tau_{ji}(t_0)) - \frac{(\sigma_i(t_0) - x_i)}{\sigma_i(t_0)}\sum_{j\in\Phi_i}(y_j - \tau_{ji}(t_0)) \\
&= \sigma_i(t_0) - \frac{(\sigma_i(t_0) - x_i)}{\sigma_i(t_0)}\sigma_i(t_0) \\
&= x_i.
\end{aligned} \tag{63}$$

For the unbiased case where $t_0 = 0$, so all $\tau_{ji}(0) = 0$,

$$S_{ji}(0) \equiv y_j - \tau_{ji}(0) = y_j \tag{64}$$

and

$$\sigma_i(0) \equiv \sum_j S_{ji}(0) = \sum_j y_j = 1. \tag{65}$$

Thus:

$$\begin{aligned}
\tau_{ji}(t) &= \tau_{ji}(0) + \alpha(t)\frac{[\sigma_i(0) - x_i]^+}{\sigma_i(0)}[y_j - \tau_{ji}(0)]^+ \\
&= \alpha(t)(1 - x_i)y_j,
\end{aligned} \tag{66}$$

$$\begin{aligned}
S_{ji}(t) &\equiv y_j - \tau_{ji}(t) \\
&= y_j - \alpha(t)(1 - x_i)y_j \\
&= y_j(1 - \alpha(t)(1 - x_i)),
\end{aligned} \tag{67}$$

and

$$S_{ji}(t) \rightarrow y_j x_i \tag{68}$$

as $t \to \infty$. By (68), when the system begins with no initial bias, the signal S_{ji} from the j^{th} F_2 node to the i^{th} F_1 node begins as y_j and converges toward the Hebbian pre- and post-synaptic correlation term $y_j x_i$.

12. 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 6a). 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 (9), and fix an F_2 code $\mathbf{y} = (y_1, y_2)$, with :

$$0 \leq y_2 \leq y_1 \leq 1. \tag{69}$$

By the F_2 normalization hypothesis (20), $y_1 + y_2 = 1$. By (11), (27), and (56), for $j = 1, 2$:

$$\frac{d}{dt}\tau_{ji} = [y_j - \tau_{ji}]^+ [\sigma_i - I_i]^+, \tag{70}$$

where, by (57),

$$\sigma_i = [y_1 - \tau_{1i}]^+ + [y_2 - \tau_{2i}]^+. \tag{71}$$

Figure 6b–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 $\sigma_i = I_i$. Where $\tau_{1i}(0) < y_1$ and $\tau_{2i}(0) < y_2$, the point $(\tau_{1i}(t), \tau_{2i}(t))$ moves along a straight line from $(\tau_{1i}(0), \tau_{2i}(0))$ toward (y_1, y_2), slowing down asymptotically as:

$$\begin{aligned} \sigma_i &= [y_1 - \tau_{1i}(t)]^+ + [y_2 - \tau_{2i}(t)]^+ \\ &= 1 - (\tau_{1i}(t) + \tau_{2i}(t)) \to I_i. \end{aligned} \tag{72}$$

Only if $I_i = 0$ does (τ_{1i}, τ_{2i}) approach (y_1, y_2). Larger thresholds τ_{ji}, which make $\sigma_i \leq I_i$, are unchanged during learning. Small I_i allow the greatest threshold changes (Figure 6b). If $I_i = 0$,

$$\tau_{ji} \to y_j \tag{73}$$

as σ_i decreases to 0.

Both thresholds grow if both are initially small. However, if one threshold is so large as to prevent $F_2 \to F_1$ 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 is no signal from the F_2 node $j = 2$ to the i^{th} F_1 node, and hence no threshold change in that path. If, then, $\tau_{1i}(0) < y_1 - I_i$, τ_{1i} increases until:

$$\sigma_i = y_1 - \tau_{1i} \to x_i = I_i. \tag{74}$$

Larger I_i values permit threshold changes only for smaller initial threshold values. In Figure 6c, τ_{2i} can change only if τ_{1i} changes as well, when both are initially small. In contrast, since 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 6d) adaptive changes can occur only if both τ_{1i} and τ_{2i} are initially small, as they are before any learning has taken place.

13. The unit of memory

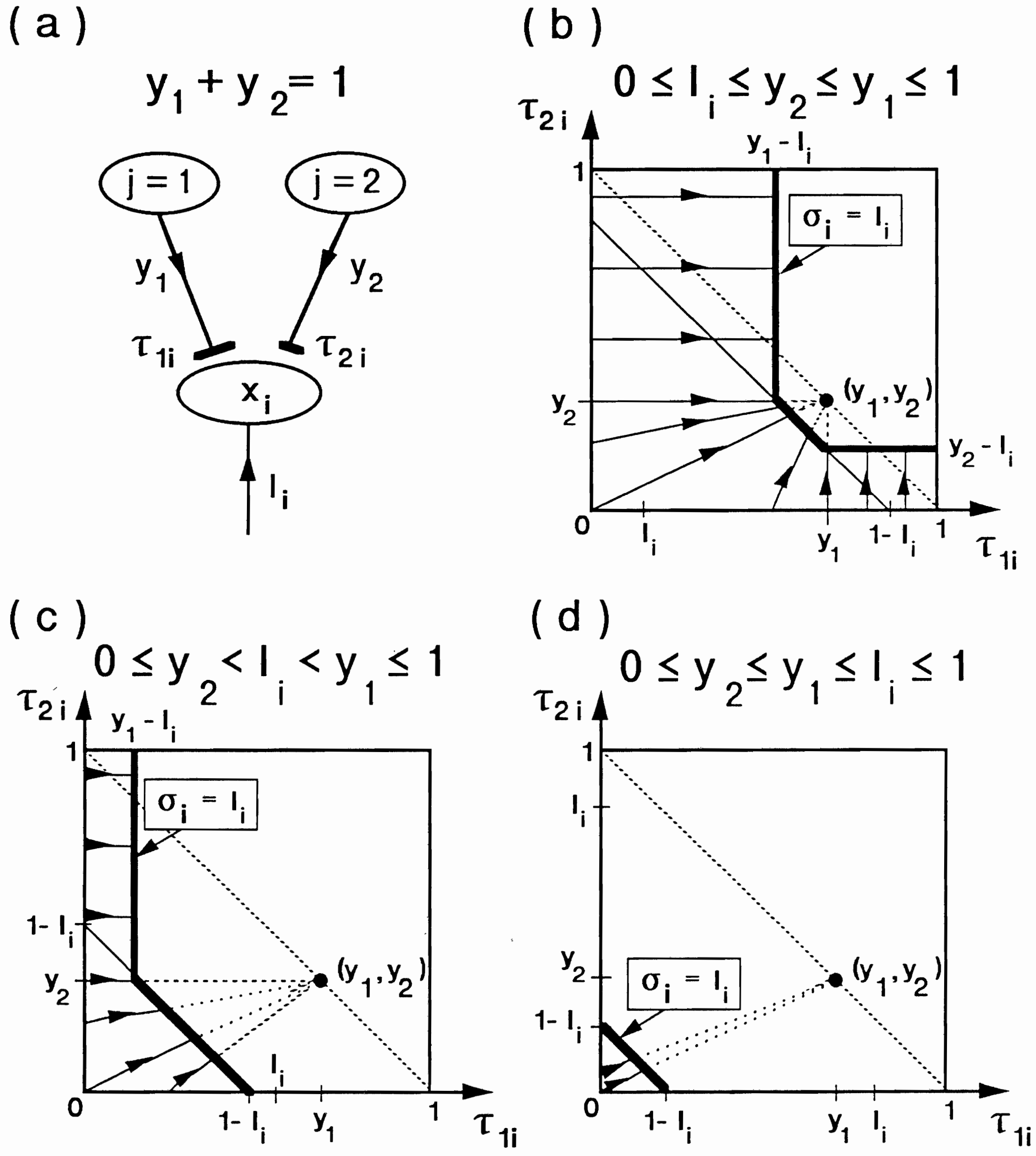

Figure 6. (a) A distributed outstar whose coding field F_2 has just two nodes ($N = 2$). For each code $\mathbf{y}$, $y_1 + y_2 = 1$, and $x_i = I_i \wedge \sigma_i$. When thresholds start out small enough, τ_{1i} and/or τ_{2i} increase toward $\{(\tau_{1i}, \tau_{2i}) : \sigma_i = I_i\}$. (b) Threshold changes are greatest for small I_i. (c) When $I_i > y_j$, the j^{th} node cannot dominate learning. Here, $I_i > y_2$, so τ_{2i} can change only when τ_{1i} also changes. (d) When I_i is large, only small thresholds can change at all.

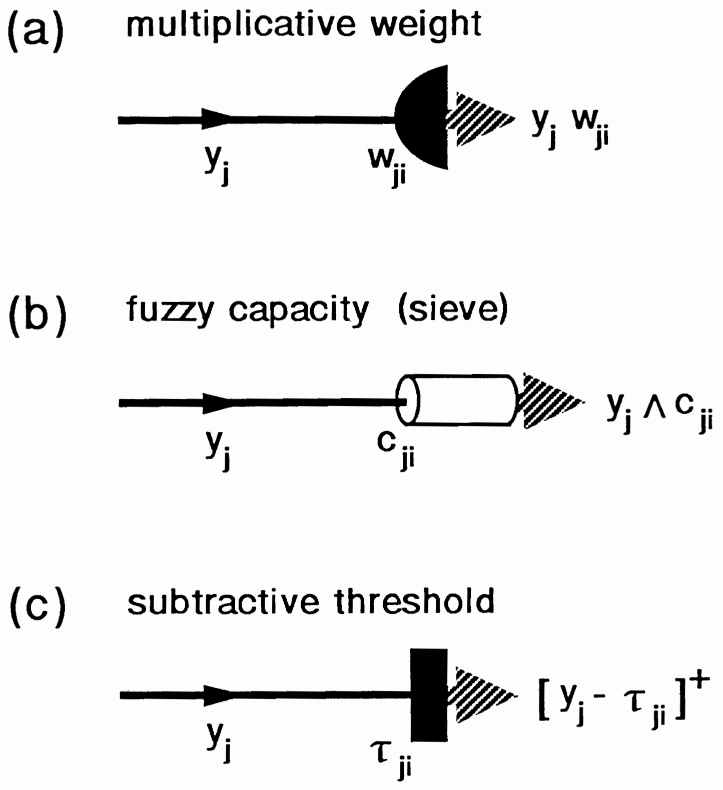

Figure 7. (a) The product rule implies a physical substrate of memory that is a multiplicative weight (McCulloch and Pitts, 1943). (b) The capacity rule implies a memory unit that is a fuzzy sieve (Zadeh, 1965). (c) The distributed outstar implies a memory unit that is a subtractive threshold.

The distributed outstar network derives from a computational analysis of stable pattern learning by distributed codes. In the distributed outstar, the adaptive threshold rule of synaptic transmission solves a catastrophic forgetting problem caused by other rules. Since each formal transmission rule corresponds to a physical theory of synaptic transmission, computational analysis implies physiological prediction. Each transmission rule assumes a physical memory unit: a multiplicative weight (Figure 7a), a fuzzy capacity, or sieve (Figure 7b), or a subtractive threshold (Figure 7c). Experiments that probe distributed coding in a living organism may be able to distinguish the three types of memory unit. Similarly, distributed outstar computations imply distinct physical realizations of optical and electronic neural networks.

REFERENCES

Carpenter, G.A. (1993). Distributed outstar learning and the rules of synaptic transmission. *Proceedings of the World Congress on Neural Networks (WCNN-93)*, **II**, 397–404.

Carpenter, G.A. (1994). A distributed outstar network for spatial pattern learning. *Neural Networks*, **7**. Technical Report CAS/CNS TR-93-036, Boston, MA: Boston University.

Carpenter, G.A. and 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 and S. Grossberg (Eds.) (1991). **Pattern Recognition by Self-Organizing Neural Networks**. Cambridge, MA: MIT Press, pp. 316–382.

Carpenter, G.A. and 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 and S. Grossberg (Eds.) (1991). **Pattern Recognition by Self-Organizing Neural Networks**. Cambridge, MA: MIT Press, pp. 398–423.

Carpenter, G.A. and 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 and S. Grossberg (Eds.) (1991). **Pattern Recognition by Self-Organizing Neural Networks**. Cambridge, MA: MIT Press, pp. 451–499.

Carpenter, G.A. and Grossberg, S. (1993). Fuzzy ARTMAP: A synthesis of neural networks and fuzzy logic for supervised categorization and nonstationary prediction. In R.R. Yager and L.A. Zadeh (Eds.). **Fuzzy Sets, Neural Networks, and Soft Computing**. New York, NY: Van Nostrand Reinhold.

Carpenter, G.A., Grossberg, S., and 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 and S. Grossberg (Eds.) (1991). **Pattern Recognition by Self-Organizing Neural Networks**. Cambridge, MA: MIT Press, pp. 503–546.

Carpenter, G.A., Grossberg, S., and Rosen, D.B. (1991a). Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, **4**, 759–771. Technical Report CAS/CNS-TR-91-015, Boston, MA: Boston University.

Carpenter, G.A., Grossberg, S., and Rosen, D.B. (1991b). A neural network realization of fuzzy ART. Technical Report CAS/CNS TR-91-021, Boston, MA: Boston University.

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*, **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**. New York: Academic Press, pp. 121–160. Reprinted in S. Grossberg

(Ed.) (1982). **Studies of Mind and Brain**. Dordrecht, Holland: D. Reidel Publishing Co., pp. 159–193.

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**. Dordrecht, Holland: D. Reidel Publishing Co., pp. 334–378.

Grossberg, S. (1974). Classical and instrumental learning by neural networks. In R. Rosen and F. Snell (Eds.), **Progress in Theoretical Biology, Volume 3**. New York: Academic Press, pp. 51–141. Reprinted in S. Grossberg (Ed.) (1982). **Studies of Mind and Brain**. Dordrecht, Holland: D. Reidel Publishing Co., pp. 68–156.

Grossberg, S. (1976). Adaptive pattern classification and universal recoding, II: Feedback, expectation, olfaction, illusions. *Biological Cybernetics*, **23**, 187–202. Reprinted in G.A. Carpenter and S. Grossberg (Eds.) (1991). **Pattern Recognition by Self-Organizing Neural Networks**. Cambridge, MA: MIT Press, pp. 283–315.

McCulloch, W.S. and 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 and E. Rosenfeld (Eds.) (1988). **Neurocomputing: Foundations of Research**. Cambridge, MA: MIT Press, pp. 18–27.

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 and E. Rosenfeld (Eds.) (1988). **Neurocomputing: Foundations of Research**. Cambridge, MA: MIT Press, pp. 92–114.

Rosenblatt, F. (1962). **Principles of Neurodynamics**. Washington, DC: Spartan Books.

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

Zadeh, L. (1965). Fuzzy sets. *Information Control*, **8**, 338-353.