Pergamon

Neural Networks, Vol. 10, No. 8, pp. 1473–1494, 1997
© 1997 Elsevier Science Ltd. All rights reserved
Printed in Great Britain
0893–6080/97 $17.00+.00

PII: S0893-6080(97)00004-X

CONTRIBUTED ARTICLE

Distributed Learning, Recognition, and Prediction by ART and ARTMAP Neural Networks

GAIL A. CARPENTER

Boston University, MA

(*Received* 1 *February* 1996; *accepted* 19 *December* 1996)

Abstract—*A class of adaptive resonance theory (ART) models for learning, recognition, and prediction with arbitrarily distributed code representations is introduced. Distributed ART neural networks combine the stable fast learning capabilities of winner-take-all ART systems with the noise tolerance and code compression capabilities of multilayer perceptrons. With a winner-take-all code, the unsupervised model dART reduces to fuzzy ART and the supervised model dARTMAP reduces to fuzzy ARTMAP. With a distributed code, these networks automatically apportion learned changes according to the degree of activation of each coding node, which permits fast as well as slow learning without catastrophic forgetting. Distributed ART models replace the traditional neural network path weight with a dynamic weight equal to the rectified difference between coding node activation and an adaptive threshold. Thresholds increase monotonically during learning according to a principle of atrophy due to disuse. However, monotonic change at the synaptic level manifests itself as bidirectional change at the dynamic level, where the result of adaptation resembles long-term potentiation (LTP) for single-pulse or low frequency test inputs but can resemble long-term depression (LTD) for higher frequency test inputs. This paradoxical behavior is traced to dual computational properties of phasic and tonic coding signal components. A parallel distributed match–reset–search process also helps stabilize memory. Without the match–reset–search system, dART becomes a type of distributed competitive learning network. © 1997 Elsevier Science Ltd.*

Keywords—Distributed ART, Adaptive resonance theory, Distributed coding, Dynamic weight, Fast learning, Competitive learning, ARTMAP, Neural network.

1. INTRODUCTION: ART, ARTMAP, AND DISTRIBUTED CODES

Adaptive resonance theory (ART) and ARTMAP neural networks feature winner-take-all competitive activation, which permits fast learning and stable coding but which causes category proliferation with noisy inputs. In contrast, multilayer perceptron models feature distributed McCulloch–Pitts activation, which enables good noise tolerance and code compression but which causes catastrophic forgetting with fast learning. This paper introduces a family of neural networks, called distributed ART models, that combine the best of these two worlds: distributed activation provides noise tolerance and code compression while new system dynamics retain stable fast learning capabilities, as follows.

1.1. ART and ARTMAP Networks

The theory of adaptive resonance began with an analysis of human cognitive information processing and stable coding in a complex input environment (Grossberg, 1976a, 1980). ART neural network models have added a series of new principles to the original theory and have realized these principles as quantitative systems that can be applied to problems of category learning, recognition, and prediction. Applications of unsupervised ART networks (Carpenter & Grossberg, 1987, 1991; Carpenter, Grossberg & Rosen, 1991b) and supervised ARTMAP networks (Carpenter, Grossberg, Markuzon, Reynolds & Rosen, 1992; Carpenter, Grossberg & Reynolds, 1991a) include a Boeing parts design retrieval system (Caudell, Smith, Escobedo & Anderson, 1994), satellite remote sensing (Baraldi & Parmiggiani, 1995; Gopal, Sklarew & Lambin, 1994), robot sensory motor control (Bachelder, Waxman & Seibert, 1993; Baloch & Waxman, 1991; Dubrawski & Crowley, 1994; Srinivasa & Sharma, 1996), robot navigation (Racz & Dubrawski, 1995), machine vision (Caudell & Healy, 1994), three-

Acknowledgements: This research was supported in part by the National Science Foundation (NSF IRI 94-01659) and the Office of Naval Research (ONR N00014-95-1-0409 and ONR N00014-95-0657).

Requests for reprints should be sent to Prof. Gail A. Carpenter, Department of Cognitive and Neural Systems, 677 Beacon Street, Boston University, Boston, MA 02215, USA; e-mail: gail@cns.bu.edu.

dimensional object recognition (Seibert & Waxman, 1992), face recognition (Seibert & Waxman, 1993), automatic target recognition (Bernardon & Carrick, 1995; Koch, Moya, Hostetler & Fogler, 1995; Waxman et al., 1995), medical imaging (Soliz & Donohoe, 1996), electrocardiogram wave recognition (Ham & Han, 1996; Suzuki, 1995), prediction of protein secondary structure (Mehta, Vij & Rabelo, 1993), strength prediction for concrete mixes (Kasperkiewicz, Racz & Dubrawski, 1995), signature verification (Murshed, Bortozzi & Sabourin, 1995), tool failure monitoring (Ly & Choi, 1994; Tarng, Li & Chen, 1994; Tse & Wang, 1994), chemical analysis from UV and IR spectra (Wienke, 1994), digital circuit design (Kalkunte, Kumar & Patnaik, 1992), frequency selective surface design for electromagnetic system devices (Christodoulou, Huang, Georgiopoulos & Liou, 1995), Chinese character recognition (Gan & Lua, 1992; Kim, Jung, Kim & Kim, 1995), and analysis of musical scores (Gjerdingen, 1990). Despite the growing number of applications, category proliferation from noisy training sets limits the useful domain of fast-learn, winner-take-all (WTA) systems such as ART or ARTMAP. On the other hand, fast learning is often essential for online adaptation to rapidly changing circumstances and for encoding rare cases and large databases.

Variants of the basic ART and ARTMAP networks have acquired some of the advantages of distributed coding while maintaining the fast learning capability. For example, ART-EMAP, which uses WTA codes for learning and distributed codes for testing, can significantly improve ARTMAP performance, especially when the size of the training set is small (Carpenter & Ross, 1993, 1995; Rubin, 1995). In medical database prediction problems, which often feature inconsistent training input predictions, ARTMAP-IC improves performance with a combination of distributed prediction, category instance counting, and a new match tracking search algorithm (Carpenter & Markuzon, 1996). A voting strategy further increases predictive accuracy by training the system several times on different orderings of an input set. Voting, instance counting, and distributed representations combine to form confidence estimates for competing predictions. However, since these and most other ART and ARTMAP variants use WTA coding during learning, they do not solve the primary problem of category proliferation with noisy training sets, unless learning is slow.

The new family of distributed ART models retains stable coding, recognition, and prediction, but allows arbitrarily distributed code representation during learning as well as performance. When the code is winner-take-all, the unsupervised dART model is computationally equivalent to fuzzy ART and the supervised dARTMAP model is equivalent to fuzzy ARTMAP. Distributed ART networks automatically apportion learned changes according to the degree of activation of each coding node. This permits fast as well as slow learning without catastrophic forgetting. Many variations of the basic dART system may be devised but, for clarity, one specific network from the larger class is developed here.

1.2. Neural Analogues of dART Network Components

Distributed ART (dART) derives primarily from a computational analysis of design principles for constructing a learning system that is fast, stable, and distributed. Nevertheless, many network elements can also be visualized as physical processes with neural interpretations. In distributed ART, the fundamental synaptic memory unit is an adaptive threshold that increases during learning according to a principle of atrophy due to disuse. A dynamic weight that depends on both the coding node activation and the adaptive threshold then replaces the fuzzy ART path weight in the dART algorithm. In contrast, the fundamental synaptic memory unit in nearly all other neural networks is assumed axiomatically to be a multiplicative weight. This view of adaptation is also prevalent in the experimental literature: "Changes in the amplitude of synaptic responses evoked by single-shock extracellular electrical stimulation of presynaptic fibres are usually considered to reflect a change in the gain of synaptic signals, and are the most frequently used measure for evaluating synaptic plasticity" (Markram & Tsodyks, 1996, p. 807). That is, when long-term potentiation (LTP), or enhanced postsynaptic response to a single test pulse, is observed, the strength, or gain, of a synapse is normally interpreted as having increased. Similarly, long-term depression (LTD) is usually thought of as a weight decrease.

While the multiplicative weight model helps explain classical LTP and LTD experiments, limitations of this hypothesis are beginning to become apparent. Describing their experiments on layer-5 pyramidal neurons in the neocortex, Markram and Tsodyks point out that the enhanced response to single-spike ($\leq$ 0.25 Hz) test probes in an LTP experiment vanishes with 23 Hz test stimuli: "Potentiation of synaptic responses therefore only occurred when the presynaptic frequency was below 20 Hz" (p. 809). In fact, the Markram and Tsodyks data [Figure 3(c), p. 809] actually show depressed postsynaptic responses to higher frequency (30 Hz and 40 Hz) test stimuli. They conclude: "The physiological implications of redistribution of efficacy are also entirely different from unconditional potentiation or depression" (p. 810).

The dynamic coding behavior of distributed ART model neurons closely resembles this paradoxical "redistribution of efficacy." In dART, adaptive thresholds increase monotonically during learning, but an increased threshold produces postsynaptic potentiation for lower frequency test inputs and postsynaptic depression for higher frequencies. These bidirectional dynamics are traced to the form of the signal that activates the dART distributed code. This signal is a function of two components with dual computational properties: a *phasic component* that depends on the transmitted input (ligand) and a *tonic component* that is independent of the current input. Both phasic and tonic components depend on the size of the adaptive threshold for the synapse and on the degree of activation of the target node (voltage). Phasic and tonic components can thus be visualized as postsynaptic membrane processes with phasic terms mediated by voltage- and ligand-gated receptors and tonic terms mediated by voltage-gated receptors (Nicholls, 1994). At each synapse, phasic and tonic terms dynamically balance one another. During adaptation, phasic terms remain constant while tonic terms may grow. Tonic components then become larger for all inputs, but phasic components become more selective. The net effect is to enhance the total coding signal subsequently sent by input components that are the same as or smaller than the one experienced during training (potentiation) but to reduce the total coding signal sent by input components that are substantially larger than those experienced during training (depression).

Analysis of the Markram and Tsodyks data illustrates how computational modeling of distributed pattern coding by neural network architectures is connected to important current questions concerning the underlying neural mechanisms of learning and memory. Phasic and tonic signals in the dART model, originally derived from a formal analysis of distributed pattern learning, demonstrate how: ''Redistribution of synaptic efficacy may therefore serve as a powerful mechanism to alter the dynamics of synaptic transmission in subtle ways and hence to alter the content rather than the gain of signals conveyed between neurons'' (Markram & Tsodyks, 1996, p. 810). The remainder of this paper will henceforth focus primarily on the design of distributed ART.

1.3. Outline

Section 2 introduces the dART architecture and formally defines dynamic weights, adaptive thresholds, and phasic and tonic signal components, and characterizes the distributed code that a given input will activate. Section 3 describes a parallel distributed match–reset–search process. Section 4 outlines the distributed outstar used for top–down dART learning and introduces the distributed instar used for bottom–up learning. Dynamics of a distributed competitive learning module are also characterized. Section 5 summarizes a dART algorithm for simulation implementation. With winner-take-all dynamics at the coding field F_2, the dART algorithm reduces to a fuzzy ART algorithm, and further reduces to an ART 1 algorithm with binary inputs. Section 6 provides a geometric representation of distributed ART and Section 7 includes numerical examples of dART activation, search, and learning. Finally, Section 8 describes distributed ARTMAP.

2. DISTRIBUTED ACTIVATION

Over the past decade, an evolving series of neural network models has progressively expanded the domain and function of ART systems. The first model, ART 1 (Carpenter & Grossberg, 1987), is an unsupervised learning system that self-organizes recognition categories for binary input patterns. Fuzzy ART (Carpenter et al., 1991b) generalizes binary ART 1 to the analog input domain, formally replacing set-theoretic intersections with fuzzy set-theoretic intersections [Figure 1(a)]. These and most other ART models use choice, or winner-take-all (WTA), dynamics at the category representation field. Distributed ART (dART) continues the series, generalizing fuzzy ART to permit arbitrarily distributed code representations [Figure 1(b)]. For continuity, dART retains fuzzy ART notation wherever possible.

2.1. dART Network Architecture

Although dART with winner-take-all coding is computationally equivalent to fuzzy ART, the dART architecture differs from the standard ART architecture. Namely, an ART input from a field F_0 passes through a matching field F_1 before activating a coding field F_2. Activity at F_2 feeds back to F_1, forming a resonant loop [Figure 1(a)]. ART networks thus encode matched F_1 patterns, rather than the F_0 inputs themselves, a key feature for code stability. With WTA coding, the matched F_1 pattern confirms the original category choice when it feeds back up to F_2. The critical code confirmation property may not persist in this architecture, however, when F_2 activation is distributed. In contrast, in the distributed ART network, the coding field F_2 receives input directly from F_0, retaining the bottom–up/top–down matching process at F_1 only to determine whether an active code meets the vigilance matching criterion [Figure 1(b)]. Nevertheless, dART dynamic weights maintain code stability when F_2 coding is winner-take-all. When the matching process is disabled by setting the vigilance parameter to 0, dART becomes a type of feedforward ART network that can also be viewed as a new type of distributed competitive learning architecture.

2.2. Activity Vectors

A dART system includes a field of nodes F_0 that represents a current input vector; a field F_2 that represents the active code; and a field F_1 that represents a matched pattern determined by bottom–up input from F_0 and top–down input from F_2. Vector $\mathbf{I} \equiv (I_1 \ldots I_i \ldots I_M)$ denotes F_0 activity, $\mathbf{x} \equiv (x_1 \ldots x_i \ldots x_M)$ denotes F_1 activity, and $\mathbf{y} \equiv (y_1 \ldots y_j \ldots y_N)$ denotes F_2 activity. Each component of $\mathbf{I}$, $\mathbf{x}$, and $\mathbf{y}$ is contained in the interval

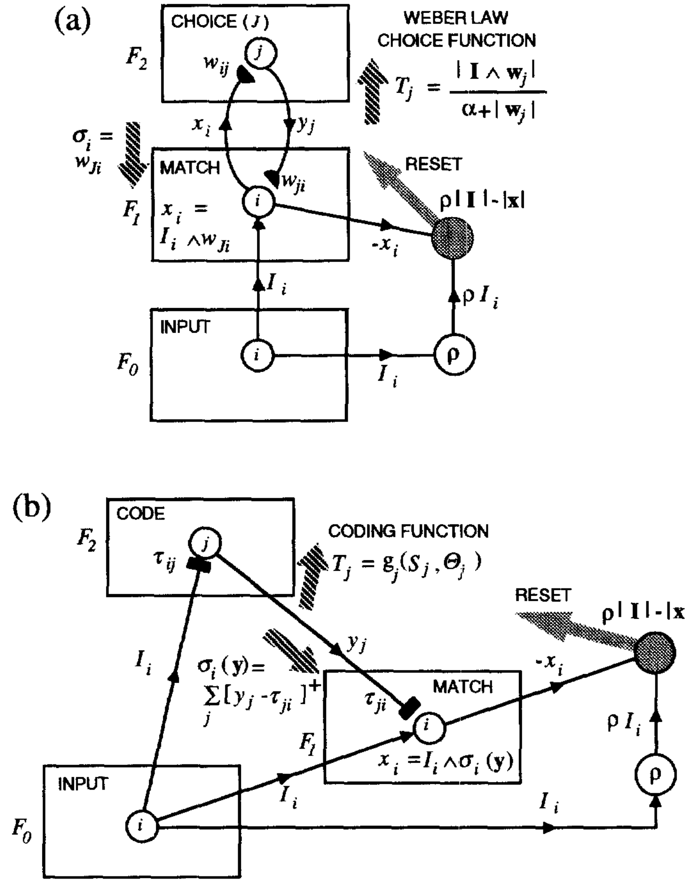

FIGURE 1. Fuzzy ART and distributed ART: (a) in fuzzy ART, the F_2 the node ($j = J$) that receives the largest input T_j from F_1 becomes active. Activity x at the field F_1 reflects the match between the bottom-up input I and the top-down input σ, which is equal to the weight vector $\mathbf{w}_J$. When x fails to meet the vigilance matching criterion, reset leaves node J refractory on the time scale of search. Refractory nodes recover on the time scale of learning; (b) like fuzzy ART, distributed ART computes a matched pattern x at F_1 and resets F_2 if x fails to meet the vigilance matching criterion. In dART, however, F_2 receives input directly from F_0. The code y, which is a function of phasic components S_j and tonic components Θ_j, may be arbitrarily distributed. The i^{th} F_1 node receives a positive signal from each F_2 node at which activity y_j exceeds an $F_2 \rightarrow F_1$ adaptive threshold τ_{ji}. With choice at F_2 and fast learning, distributed ART is computationally equivalent to fuzzy ART.

[0,1]. The number of input components (M) and the number of coding nodes (N) may be arbitrarily large. Although the matched F_1 activity vector **x** does not feed back to F_2 (Figure 1), dART still performs computations that are equivalent to those of fuzzy ART in the special case of category choice at F_2. The input **I** and the matched pattern **x** may be continuously varying functions of time t, but the code **y** acts as a content-addressable memory (CAM) that is held constant between resets by strong competition at F_2.

2.3. Dynamic Weights

In fuzzy ART the path from the i^{th} F$_1$ node to the j^{th} F_2 node contains an adaptive weight w_{ij}, and the path from the j^{th} F_2 node to the i^{th} F_1 node contains a weight w_{ji}. With fast learning, $w_{ij} \equiv w_{ji}$. Nearly all neural network models hypothesize such a weight as the unit of long-term memory (LTM). In contrast, in the distributed outstar network (Carpenter, 1994a) the unit of long-term memory is an adaptive threshold τ_{ji}. Formally,

$$\tau_{ji} \equiv 1 - w_{ji}. \qquad (1)$$

The distributed outstar signal from the j^{th} F_2 node to the i^{th} F_1 node is $[y_j - \tau_{ji}]^+$, where $[\ldots]^+$ denotes the rectification operator:

$$[\xi]^+ = \max\{\xi, 0\}. \qquad (2)$$

This path signal helps avoid catastrophic forgetting because $[y_j - \tau_{ji}]^+ = [w_{ji} - (1 - y_j)]^+ = 0$ when w_{ji} is small, unless $y_j = 1$. Other types of signals such as the product $y_j w_{ji}$ remain positive when y_j is positive, no matter how small the weight has become, leaving w_{ji} subject to erosion. When the j^{th} F_2 node is chosen, $w_{ji} = (1 - \tau_{ji}) = [y_j - \tau_{ji}]^+$.

Distributed ART takes this idea one step further, replacing each fuzzy ART weight with a *dynamic weight* that is a joint function of coding node activation and an adaptive threshold. The formal substitution:

$$w_{ji} \rightarrow [y_j - \tau_{ji}]^+ \tag{3}$$

and

$$w_{ij} \rightarrow [y_j - \tau_{ij}]^+ \tag{4}$$

is the key step in converting fuzzy ART to distributed ART. Thresholds τ_{ji} in paths from the j^{th} F_2 node to the i^{th} F_1 node adapt according to a distributed outstar learning law (Section 4.1), while thresholds τ_{ij} in paths from the i^{th} F_0 node to the j^{th} F_2 node obey a distributed instar learning law (Section 4.2). Adaptive thresholds remain in the range [0,1], starting at or near 0 and increasing monotonically during learning.

2.4. Signal Functions

For each input **I** and $j = 1 \dots N$, the total signal T_j from the dART input field F_0 to the j^{th} F_2 node is a function of the form:

$$T_j = T_j(y_j) = g_j(S_j(y_j), \Theta_j(y_j)). \tag{5}$$

For $S_j > 0$ and $\Theta_j > 0$,

$$\frac{\partial g_j}{\partial S_j} > \frac{\partial g_j}{\partial \Theta_j} > 0, \tag{6}$$

and

$$g_j(0,0) = 0. \tag{7}$$

The definition of the $F_0 \rightarrow F_2$ signal T_j at first appears to be circular: T_j determines the F_2 code **y** (Figure 1), but **y** in turn determines T_j [eqn (5)]. However, this circularity does not actually occur in distributed ART dynamics. Because the competitive field F_2 acts as a content-addressable memory, the network holds **y** constant between resets (Section 2.2). Upon reset, a large nonspecific arousal signal breaks the CAM competitive feedback loop, momentarily sending all y_j values to 1 (Section 2.5). The code **y** at any given time is therefore fully determined by the value of the signals $T_j(1)$ at the time of the previous reset. $T_j(y_j)$ represents the synaptic processes that, having survived the competition at reset, determine the dynamics of search (Section 3.3) and learning (Section 4.2) between resets. Since total F_2 activity is normalized to 1 (Section 2.5), active nodes typically represent a concentrated subset of the field's total capacity (N), which can be arbitrarily large. Correspondingly, the signal $T_j(y_j)$ between resets is on average a small fraction of the signal $T_j(1)$ at the time of reset.

In eqn (5) the *phasic* component S_j, which depends on the input **I**, is a sum:

$$S_j = S_j(y_j) = \sum_{i=1}^{M} S_{ij}(y_j). \tag{8}$$

A term $S_{ij}(y_j)$ in the sum may be visualized as a certain fraction of the membrane sites at the i^{th} synapse of the j^{th} F_2 node (Figure 2). After a new input **I** establishes an F_2 code **y** at the time of reset, phasic sites primed by the dynamic weight $[y_j - \tau_{ij}]^+$ can remain activated by the input I_i, although a number of these sites (Δ_{ij}) may be refractory, or depleted, due to their recent activation during search (Section 3). Formally,

$$S_{ij}(y_j) = \left[I_i \wedge [y_j - \tau_{ij}]^+ - \Delta_{ij}\right]^+, \tag{9}$$

where $\wedge$ represents the fuzzy intersection, or component-wise minimum:

$$(\mathbf{a} \wedge \mathbf{b})_i \equiv (a_i \wedge b_i) \equiv \min(a_i, b_i) \tag{10}$$

(Zadeh, 1965). For $y_j \epsilon [0,1]$,

$$0 \leq S_j(y_j) \leq \sum_{i=1}^{M} [y_j - \tau_{ij}]^+ \leq \sum_{i=1}^{M} y_j = My_j. \tag{11}$$

In eqn (5), the *tonic* component Θ_j is a sum:

$$\Theta_j = \Theta_j(y_j) = \sum_{i=1}^{M} \Theta_{ij}(y_j) \tag{12}$$

where:

$$\Theta_{ij}(y_j) = [y_j \wedge \tau_{ij} - \delta_{ij}]^+. \tag{13}$$

The sum $\Theta_j(y_j)$, which is independent of the input **I**, plays the role of a nodal bias term that increases during learning. Once **y** is established following a reset, a fraction of membrane sites τ_{ij} are primed by the node's activity (y_j), but recently active sites (δ_{ij}) may be refractory during search. Like $S_j(y_j)$, $\Theta_j(y_j)$ lies in the interval $[0, My_j]$ since:

$$0 \leq \Theta_j(y_j) \leq \sum_{i=1}^{M} y_j = My_j. \tag{14}$$

Refractory sites accumulate during a rapid series of resets. On the time scale of learning, the terms Δ_{ij} and δ_{ij} decay to 0.

By design, the phasic and tonic components of the F_2 input signal T_j play complementary roles in dART networks. Each phasic term is an increasing function of I_i. However, when **I** and **y** remain constant during a learning interval, the phasic terms $S_{ij}(y_j) = I_i \wedge [y_j - \tau_{ij}]^+$ and $S_{ij}(1) = I_i \wedge (1 - \tau_{ij})$ remain constant (Section 4.2). In contrast, each tonic term is, by definition, independent of **I**. However, the tonic terms $\Theta_{ij}(y_j) = y_j \wedge \tau_{ij}$ and $\Theta_{ij}(1) = \tau_{ij}$ increase during learning, when y_j is large enough. Thus by eqns (5) and (6), T_j is an increasing function of each component of **I** and T_j increases during learning.

A distributed version of the fuzzy ART choice-by-difference (CBD) function (Carpenter & Gjaja, 1994) defines one signal rule for T_j by:

$$T_j = S_j + (1 - \alpha)\Theta_j, \tag{15}$$

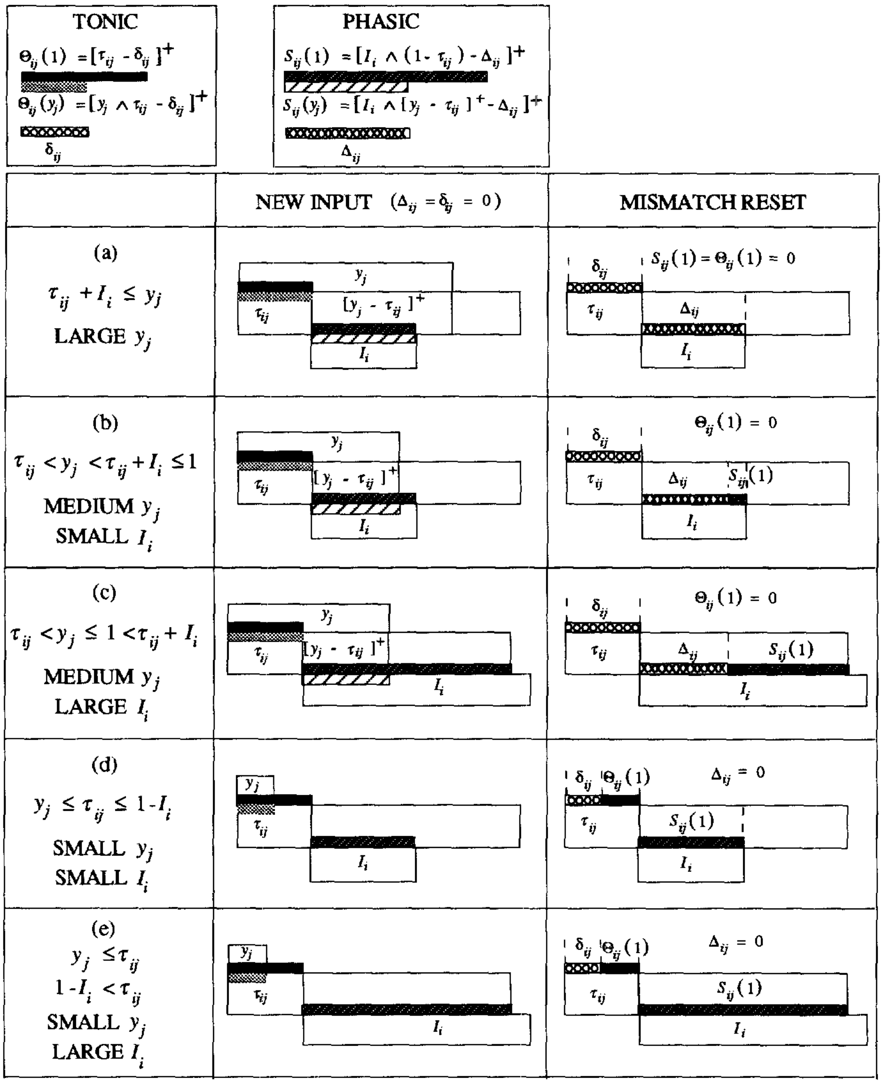
TONIC
Θ_ij(1) = [τ_ij - δ_ij]^+
Θ_ij(y_j) = [y_j ∧ τ_ij - δ_ij]^+
δ_ij
PHASIC
S_ij(1) = [I_i ∧ (1 - τ_ij) - Δ_ij]^+
S_ij(y_j) = [I_i ∧ [y_j - τ_ij]^+ - Δ_ij]^+
Δ_ij
NEW INPUT (Δ_ij = δ_ij = 0)
MISMATCH RESET
(a) τ_ij + I_i ≤ y_j LARGE y_j
(b) τ_ij < y_j < τ_ij + I_i ≤ 1 MEDIUM y_j SMALL I_i
(c) τ_ij < y_j ≤ 1 < τ_ij + I_i MEDIUM y_j LARGE I_i
(d) y_j ≤ τ_ij ≤ 1 - I_i SMALL y_j SMALL I_i
(e) y_j ≤ τ_ij 1 - I_i < τ_ij SMALL y_j LARGE I_i
y_j
τ_ij
[y_j - τ_ij]^+
I_i
δ_ij
Δ_ij
S_ij(1) = Θ_ij(1) = 0
Θ_ij(1) = 0
S_ij(1)
Θ_ij(1)
Δ_ij = 0

FIGURE 2. Visual representation of distributed instar signal components as a fraction of total membrane sites. The phasic term $S_{ij}(y_j)$ and the tonic term $\Theta_{ij}(y_j)$ depend on the adaptive threshold τ_{ij} at the i^{th} synapse of the j^{th} F_2 node. At reset, nonspecific arousal momentarily sends all $y_j \rightarrow 1$. The terms $S_{ij}(1)$ and $\Theta_{ij}(1)$ at the time of reset then determine the next code y. A given y_j value gates membrane sites, so that $S_{ij}(y_j)$ and $\Theta_{ij}(y_j)$ may be large for large y_j but must be small for small y_j. Phasic and tonic terms thus correspond to membrane processes that are gated by postsynaptic voltage (y_j), and the phasic term S_{ij} is also gated by the released presynaptic transmitter, or ligand (I_i). After mismatch reset, previously active sites Δ_{ij} (phasic) and δ_{ij} (tonic) are depleted, or refractory, and remain so on an MTM time scale. During a search, phasic and tonic terms $S_{ij}(1)$ and $\Theta_{ij}(1)$ can be large only if y_j has recently remained small.

with $0 < \alpha < 1$. Like S_j and Θ_j, the CBD signal function $T_j \in [0,My_j]$since:

$$0 \le T_j(y_j) = S_j(y_j) + (1-\alpha)\Theta_j(y_j) \le S_j(y_j) + \Theta_j(y_j) \qquad (16)$$

$$\le \sum_{i=1}^{M} I_i \wedge [y_j - \tau_{ij}]^+ + \sum_{i=1}^{M} y_j \wedge \tau_{ij}$$

$$\le \sum_{i=1}^{M} \left([y_j - \tau_{ij}]^+ + y_j \wedge \tau_{ij}\right)$$

$$= \sum_{i=1}^{M} \left((y_j - y_j \wedge \tau_{ij}) + y_j \wedge \tau_{ij}\right) = \sum_{i=1}^{M} y_j = My_j.$$

A distributed version of the Weber law signal function (Carpenter & Grossberg, 1987) defines a different signal rule for T_j by:

$$T_j = \frac{S_j}{\alpha + My_j - \Theta_j}, \qquad (17)$$

with $\alpha > 0$. For the Weber law coding function [eqn (17)], $T_j \in [0,1)$since:

$$0 \le T_j = \frac{S_j}{\alpha + My_j - \Theta_j} \qquad (18)$$

$$\le \frac{\sum_{i=1}^{M} I_i \wedge [y_j - \tau_{ij}]^+}{\alpha + My_j - \sum_{i=1}^{M} y_j \wedge \tau_{ij}} \le \frac{\sum_{i=1}^{M} [y_j - \tau_{ij}]^+}{\alpha + \sum_{i=1}^{M} [y_j - \tau_{ij}]^+}$$

$$\le \frac{My_j}{\alpha + My_j} < 1.$$

In the case where $y_j \equiv 1$, $\Delta_{ij} = \delta_{ij} = 0$, and $w_{ij} \equiv (1 - \tau_{ij})$:

$$S_j = S_j(1) = |\mathbf{I} \wedge \mathbf{w}_j| \tag{19}$$

and

$$\Theta_j = \Theta_j(1) = (M - |\mathbf{w}_j|), \tag{20}$$

where $|\ldots|$ represents the city block norm. In this case the distributed choice-by-difference eqn (15) reduces to:

$$T_j = T_j(1) = |\mathbf{I} \wedge \mathbf{w}_j| + (1 - \alpha)(M - |\mathbf{w}_j|), \tag{21}$$

which is equivalent to the fuzzy ART choice-by-difference function. The distributed Weber law eqn (17) reduces to:

$$T_j = T_j(1) = \frac{|\mathbf{I} \wedge \mathbf{w}_j|}{\alpha + |\mathbf{w}_j|}, \tag{22}$$

which is equivalent to the Weber law choice rules originally used in fuzzy ART and, when **I** is binary, ART 1.

2.5. Code Representation

In distributed ART networks, activity $\mathbf{y} = (y_1 \ldots y_j \ldots y_N)$ at a competitive coding field F_2 is stored as a content-addressable memory. An algorithm that approximates the dynamics of strong competition postulates that external inputs initially determine **y**, but then internal feedback holds **y** constant until F_2 is actively reset. Except during reset, **y** is normalized:

$$|\mathbf{y}| \equiv \sum_{j=1}^{N} y_j = 1. \tag{23}$$

In ART models, F_2 reset occurs when the bottom–up/top–down matched pattern **x** at F_1 fails to meet a matching criterion defined by a vigilance parameter ρ. Reset is effected by a large nonspecific arousal signal. In the dART model, reset momentarily sends all y_j to 1 at a time $t = r$. This allows the values $T_1(1)|_{t=r} \ldots T_N(1)|_{t=r}$ to determine which **y** will be established next. Until the next reset,

$$y_j = f_j(T_1(1) \ldots T_N(1))|_{t=r}. \tag{24}$$

Realizing F_2 as an on-center off-surround shunting competitive network suggests the hypothesis:

$$\frac{\partial f_j}{\partial T_j} \ge 0. \tag{25}$$

One class of functions that satisfy this hypothesis sets:

$$y_j = \begin{cases} \dfrac{f(T_j(1))}{\sum_{\lambda \in \Lambda} f(T_\lambda(1))} & \text{if } j \in \Lambda \\ 0 & \text{if } j \notin \Lambda \end{cases} \tag{26}$$

where Λ is a subset of $\{1 \ldots N\}$ such that $T_J \ge T_j$ for $J \in \Lambda$ and $j \notin \Lambda$; and where $f(0) \ge 0$ and $f'(\xi) \ge 0$ for $\xi > 0$. Grossberg (1976b) used a similar class of functions to approximate the dynamics of on-center off-surround shunting competitive networks. The index subset Λ might be the indices of T_j values that are greater than or equal to the collective average (above-average-T_j rule); or Λ might be the indices of the Q largest T_j values (Q-max rule). Setting $Q = 1$ corresponds to choice, or winner-take-all, coding and setting $Q = N$ makes all y_j proportional to $f(T_j(1))$. The function f might realize a power law, with:

$$f(\xi) = \xi^p \tag{27}$$

for $p > 0$. Setting $p = 1$ makes y_j proportional to $T_j(1)$ for $j \in \Lambda$, and increasing the power p models progressively stronger internal network competition, producing increasingly compressed F_2 codes. In the limit as $p \to \infty$, the system [eqns (26) and (27)] converges to the choice rule. Other types of coding fields could, for example, represent cooperative or spatially defined interactions as well as competition. Compared to ART and ARTMAP networks, where the coding rule is fixed, applications of dART and dARTMAP networks typically require comparative studies to help choose rules that give the best performance in particular cases.

3. DISTRIBUTED SEARCH

The distributed ART match–reset–search process is similar to that of other ART networks. When an F_2 code **y** becomes active, the activity pattern **x** at F_1 represents a match between the current bottom–up input **I** and a top–down input $\sigma(\mathbf{y})$. If these inputs fail to meet the vigilance matching criterion, a nonspecific reset signal shuts off the code **y**. Reset also leaves an enduring trace of **y**, or the network would simply reactivate the same code.

The search process plays a variety of roles in ART and ARTMAP systems. Since F_2 is typically a strongly competitive network, active reset of a stored code is needed for each new input to select a code that is not severely distorted by the previous steady-state at F_2. An *input reset* allows an input to register its own code when it fails to match an active top–down signal $\sigma(\mathbf{y})$.

Alternatively, a novelty signal can automatically trigger a reset when a new input is presented. Input resets segment a continuously varying input $\mathbf{I}(t)$ with a discrete series of recognition codes $\mathbf{y}^{(1)},\mathbf{y}^{(2)},\ldots$. While one code remains active, the subset of input features active at F_1 represents a focus of attention. Reset defines the boundary between one attended feature set and the next.

Search also helps to stabilize memory. Immediately after an input activates a code, a *mismatch reset* will quickly shut off $\mathbf{y}$ if it fails to meet the vigilance matching criterion. Since reset is rapid on the time scale of learning (LTM), an outlier that incorrectly activates a learned code does not disrupt memory. Traces of prior resets should endure on the time scale of short-term memory (STM) and search but should fade on the time scale of learning, since a reset code that was incorrect for one input may be correct for the next. Traces of search are thus a type of medium-term memory (MTM).

Even if $\mathbf{I}$ and $\mathbf{y}$ are constant and $\mathbf{x}$ meets the matching criterion, an increase in the vigilance parameter ρ can trigger search. Such a *vigilance reset* corresponds to increased "attentiveness" due, for example, to a prediction made by $\mathbf{y}$ having led to an error. In fact, when an ARTMAP network makes a predictive error during training, the match tracking process raises vigilance until the matching criterion fails, thus triggering a vigilance reset and search. In ARTMAP the vigilance parameter therefore becomes an internally controlled variable that may increase on the MTM time scale but that relaxes to a baseline vigilance level ($\bar{\rho}$) on the LTM time scale. Finally, reset waves might also refresh F_2 periodically, to keep the system from locking into a fixed state even if vigilance is low.

3.1. Match Representation

While $\mathbf{y}$ is fixed between resets, the total input σ_i from F_2 to the i^{th} F_1 node equals the sum of dynamic weights projecting to that node. That is:

$$\sigma_i = \sigma_i(\mathbf{y}) = \sum_{j=1}^{N} \left[y_j - \tau_{ji}\right]^+, \tag{28}$$

where $\tau_{ji} \in [0,1]$ is an adaptive threshold that starts at 0 and may increase during distributed outstar learning (Section 4.1). Since $\sum_j y_j = 1, \sigma_i \in [0,1]$. Activity $\mathbf{x}$ at F_1 then equals the fuzzy intersection of $\mathbf{I}$ and $\sigma(\mathbf{y})$, so:

$$x_i = I_i \wedge \sigma_i(\mathbf{y}) \tag{29}$$

for $i = 1\ldots M$. Signals from F_2 thereby prime F_1 in the sense that $\sigma_i(\mathbf{y})$ imposes an upper bound on inputs I_i that can be fully represented at the i^{th} F_1 node.

3.2. Resonance or Reset

Resonance occurs if the matched pattern $\mathbf{I} \wedge \sigma(\mathbf{y})$ meets the vigilance criterion:

$$\frac{|\mathbf{x}|}{|\mathbf{I}|} = \frac{|\mathbf{I} \wedge \sigma(\mathbf{y})|}{|\mathbf{I}|} \geq \rho; \tag{30}$$

that is, resonance occurs if

$$|\mathbf{x}| = |\mathbf{I} \wedge \sigma(\mathbf{y})| \geq \rho|\mathbf{I}|. \tag{31}$$

Learning then ensues, as defined below. During a learning interval, $\mathbf{y}$ remains constant but the input $\mathbf{I}(t)$ and the vigilance parameter ρ may vary continuously, as long as the network continues to meet the matching criterion.

Mismatch reset occurs if:

$$\frac{|\mathbf{I} \wedge \sigma(\mathbf{y})|}{|\mathbf{I}|} < \rho; \tag{32}$$

that is, if:

$$|\mathbf{x}| = |\mathbf{I} \wedge \sigma(\mathbf{y})| < \rho|\mathbf{I}|. \tag{33}$$

A nonspecific signal to F_2 then momentarily resets all y_j to 1, until the signal vector $\mathbf{T}$ establishes a new code $\mathbf{y}$ (Section 2.5). The search process must be rapid, so that no significant learning can occur with an incorrect code. Mismatch reset must also selectively bias the network against previously active codes or $\mathbf{T}$, the same as before, will reactivate the reset code.

3.3. Medium-Term Memory

When the F_2 code makes a choice, reset needs simply to deactivate the previously active node J for the duration of the MTM time scale. When $\mathbf{y}$ is distributed, a graded bias against the j^{th} node needs to reflect how large y_j has been in previously reset codes, so that highly active nodes can give way to nodes that originally received smaller inputs. Figure 3 shows how such a parallel search process can explore various F_2 code combinations until one is found that satisfies the vigilance criterion. During a rapid series of mismatch–reset events, refractory sites accumulate [Figure 3(a)–(c)]. During a learning interval, refractory sites recover [Figure 3(d)].

Distributed ART realizes the search process by assuming that, when a code $\mathbf{y}$ is active, sites corresponding to the phasic component $S_j(y_j)$ [eqn (8)] and the tonic component $\Theta_j(y_j)$ [eqn (12)] become refractory on the MTM time scale. On the time scale of search,

$$\frac{d}{dt}\Delta_{ij} \approx S_{ij}(y_j) = \left[I_i \wedge \left[y_j - \tau_{ij}\right]^+ - \Delta_{ij}\right]^+ \tag{34}$$

and:

$$\frac{d}{dt}\delta_{ij} \approx \Theta_{ij}(y_j) = \left[y_j \wedge \tau_{ij} - \delta_{ij}\right]^+. \tag{35}$$

Each term $S_{ij}(y_j)$ [eqn (9)] and $\Theta_{ij}(y_j)$ [eqn (13)] then quickly converges to 0. When the next reset occurs, $S_j(1)$ and $\Theta_j(1)$ are reduced by the previous quantities $S_j(y_j)$ and $\Theta_j(y_j)$ (Figure 2). By eqn (6), $T_j(1)$ is also reduced; with distributed choice-by-difference, $T_j(1)$ is reduced by the previous quantity $T_j(y_j)$. Nodes where y_j is

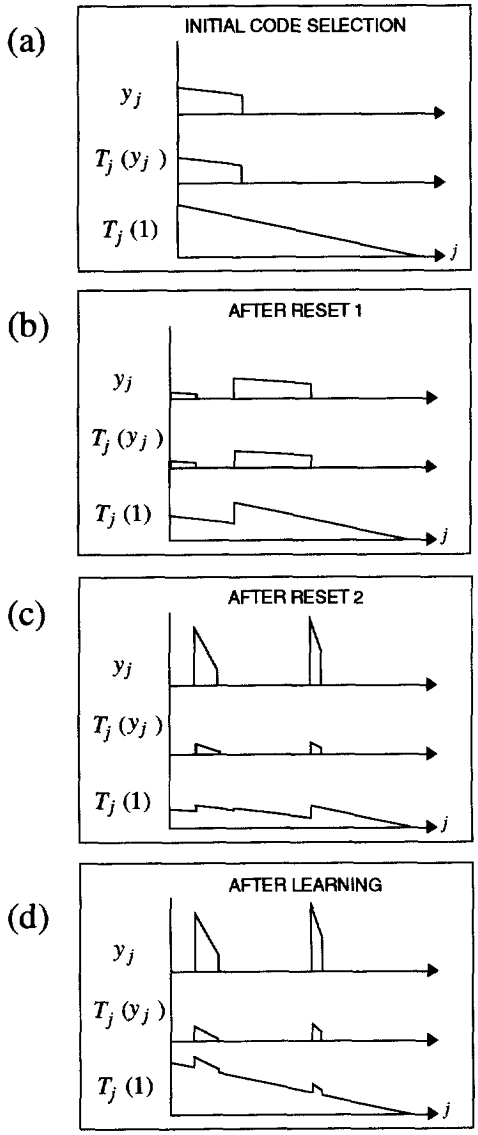

FIGURE 3. Parallel distributed search, with the F_2 code y_j proportional to $T_j(1)$ for $j \in \Lambda \subseteq \{1 \ldots N\}$ and a choice-by-difference signal rule: (a) $T_j(y_j) = 0$ for $j \notin \Lambda$; (b) after reset, $T_j(1)$ is diminished by the previous value of $T_j(y_j)$. A new set Λ of F_2 nodes where $T_j(1)$ is maximal leads to a new active code y; (c) following another reset on the time scale of search, $T_j(1)$ is further reduced by the previous value of $T_j(y_j)$; (d) refractory sites recover on the time scale of learning, so $T_j(1)$ reverts to its original value at inactive sites while $T_j(1)$ may increase where $y_j > 0$. These values of $T_j(1)$ would determine the next code y if another reset should then occur with the same I due, say, to a sudden increase in vigilance.

large tend to have the largest signals and hence the greatest reduction of the subsequent signals after a reset. When $y_j = 0$, $S_j = \Theta_j = T_j = 0$, so the signal $T_j(1)$ at the next reset will be the same as it was before.

Since recovery is slow on the time scale of search, across a rapid series of resets the phasic depletion term Δ_{ij} [eqn (9)] is approximately equal to the largest value $I_i \wedge [y_j - \tau_{ij}]^+$ has recently attained. The phasic term $S_{ij}(y_j)$ can then be positive for a new code y only if y_j is larger than it has yet been during the search. Similarly, the tonic depletion term δ_{ij} [eqn (13)] is approximately equal to the largest value $y_j \wedge \tau_{ij}$ has recently attained. Refractory sites recover on the time scale of learning. For a search where code selection is unbiased by the previous choice, the model assumes that Δ_{ij} and δ_{ij} converge to 0 during learning. When F_2 makes a choice, with $y_J = 1$, $S_J(1)$ and $\Theta_J(1)$ are reduced to 0 as $\Delta_{iJ} \rightarrow I_i \wedge (1 - \tau_{iJ})$ and $\delta_{iJ} \rightarrow \tau_{iJ}$ during search. Since $g_J(0{,}0) = 0$ [eqn (7)], $T_J(1)$ is then also reduced to 0 until it can recover on the time scale of learning.

4. DISTRIBUTED LEARNING

Catastrophic forgetting is a problem faced by all neural networks with distributed activation especially in the fast-learn limit where LTM variables go to asymptote with each input presentation. The instar and outstar learning laws used in previous ART networks would cause catastrophic forgetting if transferred to a network with a distributed code y. Stable distributed coding with fast learning requires internal or external control of the learned changes that one input can induce.

The distributed outstar (Carpenter, 1994a) solves the catastrophic forgetting problem for learning in paths that originate from the coding field F_2. The distributed instar, introduced here, solves the problem in paths that project to F_2. During distributed outstar learning, the total signal from the coding field to a target node can only decrease, by a principle of atrophy due to disuse. During distributed instar learning, the total signal to a target coding node can only increase, as the tonic component of the signal increases while the phasic component remains constant for a given input. Both learning laws bound the total learned change any one input can impose upon the system.

4.1. Distributed Outstar Learning

Dynamic weights in paths that originate from an F_2 coding node adapt according to a principle of *atrophy due to disuse*. The total top–down priming signal $\sigma_i(\mathbf{y})$ to the i^{th} F_1 node equals the sum of dynamic weights projecting to that node [eqn (28)]. During distributed outstar learning, each signal $\sigma_i(\mathbf{y})$ that exceeds the input I_i shrinks until it just "covers" I_i. Each dynamic weight $[y_j - \tau_{ji}]^+$ falls by an amount that depends upon its contribution to $\sigma_i(\mathbf{y})$ as the threshold τ_{ji} rises according to the equation:

$$\frac{d}{dt}\tau_{ji} = [y_j - \tau_{ji}]^+ \left(\sigma_i(\mathbf{y}) - x_i\right) \qquad (36)$$

$$= [y_j - \tau_{ji}]^+ \left(\sigma_i(\mathbf{y}) - I_i \wedge \sigma_i(\mathbf{y})\right)$$

$$= [y_j - \tau_{ji}]^+ [\sigma_i(\mathbf{y}) - I_i]^+,$$

by eqn (29). Initially, $\tau_{ji}(0) = 0$. By eqn (36), the sum of all thresholds to the i^{th} F_1 node increases according to:

$$\frac{d}{dt}\sum_{j=1}^{N}\tau_{ji}=\sum_{j=1}^{N}\left[y_j-\tau_{ji}\right]^{+}\left(\sigma_i(\mathbf{y})-x_i\right)=\sigma_i(\mathbf{y})\left(\sigma_i(\mathbf{y})-x_i\right) \tag{37}$$

$$=\sigma_i(\mathbf{y})\left(\sigma_i(\mathbf{y})-I_i\wedge\sigma_i(\mathbf{y})\right)=\sigma_i(\mathbf{y})\left[\sigma_i(\mathbf{y})-I_i\right]^{+}.$$

As long as I_i remains constant,

$$\sigma_i(\mathbf{y})\downarrow I_i\wedge\sigma_i(\mathbf{y})|_{t=r}, \tag{38}$$

where $t = r$ at the time of the previous reset. That is, either $\sigma_i(\mathbf{y})$ decreases toward I_i by atrophy due to disuse; or $\sigma_i(\mathbf{y})$ is smaller than I_i to begin with and so remains constant until the next reset. Activity $\mathbf{x} = \mathbf{I} \wedge \sigma(\mathbf{y})$ at the matching field F_1 thus remains constant during learning, as long as $\mathbf{I}$ and $\mathbf{y}$ remain constant.

The distributed outstar equation is simple enough to be solved directly, and its solutions are piecewise linear. If $\mathbf{I}$ and $\mathbf{y}$ remain constant during a time interval $[r,t]$, then:

$$\tau_{ji}(t)=\tau_{ji}^{\text{old}}+\phi(t)\frac{\left[\sigma_i^{\text{old}}-I_i\right]^{+}}{\sigma_i^{\text{old}}}\left[y_j-\tau_{ji}^{\text{old}}\right]^{+} \tag{39}$$

where $\tau_{ji}^{\text{old}} \equiv \tau_{ji}(r)$ and $\sigma_i^{\text{old}} \equiv \sigma_i(\mathbf{y})|_{t=r}$, and where $\phi(t)$ is an exponential that goes from 0 to 1 as t goes from r to ∞ (Carpenter, 1994b). For input presentations of fixed duration, the $F_2 \rightarrow F_1$ threshold τ_{ji} increases during learning from τ_{ji}^{old} to τ_{ji}^{new}, where:

$$\tau_{ji}^{\text{new}}=\tau_{ji}^{\text{old}}+\beta\frac{\left[\sigma_i^{\text{old}}-I_i\right]^{+}}{\sigma_i^{\text{old}}}\left[y_j-\tau_{ji}^{\text{old}}\right]^{+} \tag{40}$$

for a learning rate parameter $\beta \in [0,1]$. Setting $\beta = 1$ gives the fast-learn limit, where all variables reach asymptote during each input presentation. Eqn (39) also provides a formula for the dynamic weight $[y_j - \tau_{ji}(t)]^+$, which decreases during learning so that:

$$\left[y_j-\tau_{ji}(t)\right]^{+}\downarrow\left[y_j-\tau_{ji}^{\text{old}}\right]^{+}\frac{I_i\wedge\sigma_i^{\text{old}}}{\sigma_i^{\text{old}}}. \tag{41}$$

With category choice at F_2 and fast learning, the distributed outstar reduces to the fuzzy ART outstar, as follows. In the original outstar (Grossberg, 1968, 1970), weights w_{ji} in paths from a source node with activity y_j track target node activities x_i. The specific outstar equation used in fuzzy ART is:

$$\frac{d}{dt}w_{ji}=y_j\left(x_i-w_{ji}\right). \tag{42}$$

In fuzzy ART, with $y_J = 1$ at the chosen F_2 node, $x_i = I_i \wedge w_{Ji}$ at the i^{th} F_1 node. Initially, all $w_{ji}(0) = 1$, and top-down weights w_{ji} remain constant for $j \neq J$. For $j = J$, weights decrease by outstar learning:

$$\frac{d}{dt}w_{Ji}=\left(x_i-w_{Ji}\right)=\left(I_i\wedge w_{Ji}-w_{Ji}\right) \tag{43}$$

$$=-\left(w_{Ji}-I_i\wedge w_{Ji}\right)=-\left[w_{Ji}-I_i\right]^{+}.$$

Correspondingly, in the distributed outstar [eqn (36)] with the code $\mathbf{y}$ representing choice at F_2, $\frac{d}{dt}\tau_{ji} = 0$ for $j \neq J$.

For $j = J$,

$$\frac{d}{dt}\tau_{Ji}=\left[y_J-\tau_{Ji}\right]^{+}\left[\sigma_i(\mathbf{y})-I_i\right]^{+} \tag{44}$$

$$=\left[y_J-\tau_{Ji}\right]^{+}\left[\left[y_J-\tau_{Ji}\right]^{+}-I_i\right]^{+}$$

$$=\left(1-\tau_{Ji}\right)\left[\left(1-\tau_{Ji}\right)-I_i\right]^{+}.$$

Setting $w_{ji} \equiv (1 - \tau_{ji})$ [eqn (1)] converts eqn (44) into:

$$\frac{d}{dt}w_{Ji}=-w_{Ji}[w_{Ji}-I_i]^{+}, \tag{45}$$

with $\frac{d}{dt}(w_{ji}) = 0$ for $j \neq J$. Thus, except for the convergence rate, the distributed outstar [eqn (36)] with choice at F_2 reduces to the fuzzy ART outstar [eqn (42)]. With fast learning, the two algorithms are equivalent.

In the fuzzy ART outstar, for fixed $\mathbf{I}$ and a chosen node J, the total change in the set of weights from F_2 to the i^{th} F_1 node is bounded above by $1 - I_i$:

$$\sum_{j=1}^{N}|\Delta w_{ji}|=\left[w_{Ji}^{\text{old}}-I_i\right]^{+}\leq 1-I_i, \tag{46}$$

where $w_{Ji}^{\text{old}} \equiv w_{Ji}(r)$. In the distributed outstar, the same bound applies, with $\mathbf{y}$ arbitrarily distributed across F_2:

$$\sum_{j=1}^{N}|\Delta\tau_{ji}(t)|=\sum_{j=1}^{N}\phi(t)\frac{\left[\sigma_i^{\text{old}}-I_i\right]^{+}}{\sigma_i^{\text{old}}}\left[y_j-\tau_{ji}^{\text{old}}\right]^{+} \tag{47}$$

$$\leq\frac{\left[\sigma_i^{\text{old}}-I_i\right]^{+}}{\sigma_i^{\text{old}}}\sum_{j=1}^{N}\left[y_j-\tau_{ji}^{\text{old}}\right]^{+}=\left[\sigma_i^{\text{old}}-I_i\right]^{+}\leq 1-I_i.$$

Thus, distributed outstar learning preserves dynamic range and avoids catastrophic forgetting.

4.2. Distributed Instar Learning

Distributed instar learning is designed to enhance the competitive advantage of highly active coding nodes with respect to the current input. At the same time, learning makes these nodes more selective, so that different inputs will tend to activate distinct codes. During distributed instar learning a large dynamic weight $[y_j - \tau_{ij}]^+$ decreases toward a smaller input I_i according to the equation:

$$\frac{d}{dt}\tau_{ij}=\left[y_j-\tau_{ij}-I_i\right]^{+} \tag{48}$$

$$=\left[\left[y_j-\tau_{ij}\right]^{+}-I_i\right]^{+}$$

$$=\left(\left[y_j-\tau_{ij}\right]^{+}-I_i\wedge\left[y_j-\tau_{ij}^{\text{old}}\right]^{+}\right),$$

where $\tau_{ij}^{\text{old}} \equiv \tau_{ij}(r)$, at the time of the previous reset. Initially:

$$\tau_{ij}(0)=\eta_{ij}=0^{+}, \tag{49}$$

where the values η_{ij} are small random numbers needed to break the tie among the first F_2 inputs $T_j(1)$. As long as $\mathbf{I}$ and $\mathbf{y}$ remain constant, the threshold τ_{ij} increases:

$$\tau_{ij} \uparrow (y_j - I_i) \vee \tau_{ij}^{\text{old}}, \qquad (50)$$

where $\vee$ represents the fuzzy union, or component-wise maximum:

$$(\mathbf{a} \vee \mathbf{b})_i \equiv (a_i \vee b_i) \equiv \max(a_i, b_i) \qquad (51)$$

(Zadeh, 1965). As τ_{ij} increases, the dynamic weight $[y_j - \tau_{ij}]^+$ decreases:

$$[y_j - \tau_{ij}]^+ \downarrow I_i \wedge [y_j - \tau_{ij}^{\text{old}}]^+. \qquad (52)$$

Solving eqn (48) gives:

$$\tau_{ij}(t) = \tau_{ij}^{\text{old}} + \phi(t)[y_j - \tau_{ij}^{\text{old}} - I_i]^+ \qquad (53)$$

$$= \begin{cases} \tau_{ij}^{\text{old}} & \text{if } \tau_{ij}^{\text{old}} \geq (y_j - I_i) \\ (1 - \phi(t))\tau_{ij}^{\text{old}} + \phi(t)(y_j - I_i) & \text{if } \tau_{ij}^{\text{old}} < (y_j - I_i) \end{cases}$$

where $\phi(t)$ is an exponential that goes from 0 to 1 as t goes from r to ∞. In addition, the maximum total increase across all the MN thresholds during one input presentation is bounded above by M:

$$\sum_{i=1}^{M} \sum_{j=1}^{N} |\Delta\tau_{ij}| \leq \sum_{i=1}^{M} \sum_{j=1}^{N} [y_j - \tau_{ij}^{\text{old}} - I_i]^+ \qquad (54)$$

$$\leq \sum_{i=1}^{M} \sum_{j=1}^{N} [y_j - I_i]^+ \leq \sum_{i=1}^{M} \sum_{j=1}^{N} y_j = M.$$

For input presentations of fixed duration, τ_{ij} increases from τ_{ij}^{old} to τ_{ij}^{new} where:

$$\tau_{ij}^{\text{new}} = (1 - \beta)\tau_{ij}^{\text{old}} + \beta(y_j - I_i) \vee \tau_{ij}^{\text{old}} \qquad (55)$$

$$= \tau_{ij}^{\text{old}} + \beta[y_j - \tau_{ij}^{\text{old}} - I_i]^+$$

for the learning rate parameter $\beta \in [0,1]$. In the fast-learn limit, $\beta = 1$.

Note that, by eqns (9) and (13), $S_{ij}(y_j) = I_i \wedge [y_j - \tau_{ij}]^+$ and $\Theta_{ij}(y_j) = y_j \wedge \tau_{ij}$ during learning, since then $\Delta_{ij} = \delta_{ij} = 0$. The distributed instar learning law can thus be written in terms of the phasic and tonic signals, since:

$$\frac{d}{dt}\tau_{ij} = [[y_j - \tau_{ij}]^+ - I_i]^+ \qquad (56)$$

$$= [y_j - \tau_{ij}]^+ - I_i \wedge [y_j - \tau_{ij}]^+$$

$$= y_j - y_j \wedge \tau_{ij} - I_i \wedge [y_j - \tau_{ij}]^+$$

$$= y_j - \Theta_{ij}(y_j) - S_{ij}(y_j).$$

The term $S_{ij}(y_j)$ can be thought to represent a set of synaptic sites that are phasically activated by the input I_i, while $\Theta_{ij}(y_j)$ represents sites that are tonically activated, independent of I_i. By eqn (6), a phasically active site makes a larger contribution to the overall signal than does a tonically active site. However, the term $[y_j - S_{ij}(y_j)]$ then represents "disused" sites that are primed by postsynaptic activation y_j but are not phasically activated by the current input. During learning, τ_{ij} remains constant if the j^{th} node is relatively inactive ($y_j \leq \tau_{ij} + I_i$). Otherwise, $\Theta_{ij}(y_j)$ increases toward $[y_j - S_{ij}(y_j)]$ as disused phasic sites revert to tonic sites.

With category choice at F_2, the distributed instar reduces to the fuzzy ART instar, as follows. In the original instar (Grossberg, 1972), weights w_{ij} in paths projecting to an active target node j track activity x_i in the incoming paths. The specific instar equation used in fuzzy ART is:

$$\frac{d}{dt}w_{ij} = y_j(x_i - w_{ij}). \qquad (57)$$

In fuzzy ART, the path signal from F_1 is $x_i = I_i \wedge w_{Ji}$, where $y_J = 1$ at the chosen F_2 node. With fast learning, $w_{ij} = w_{ji}$ except initially, when w_{ij} $(0) = 1 - \eta_{ij} = 1^-$. Bottom–up weights w_{ij} remain constant for $j \neq J$. For $j = J$, weights decrease by instar learning:

$$\frac{d}{dt}w_{iJ} = (x_i - w_{iJ}) = (I_i \wedge w_{Ji} - w_{iJ}) \qquad (58)$$

$$= -(w_{iJ} - I_i \wedge w_{iJ}) = -[w_{iJ} - I_i]^+.$$

Correspondingly, setting $w_{ij} \equiv (1 - \tau_{ij})$ in the distributed instar [eqn (48)] with choice at F_2 gives:

$$\frac{d}{dt}w_{iJ} = -\frac{d}{dt}\tau_{iJ} = -[y_J - \tau_{iJ} - I_i]^+ \qquad (59)$$

$$= -[1 - \tau_{iJ} - I_i]^+ = -[w_{iJ} - I_i]^+$$

and $\frac{d}{dt}(w_{ij}) = 0$ for $j \neq J$. Thus, the distributed instar with choice at F_2 and fast learning reduces to the fuzzy ART instar [eqn (57)].

4.3. Distributed Competitive Learning

In a competitive learning network (Grossberg, 1972, 1976b; Malsburg, 1973) inputs I_i filtered through adaptive pathways produce activations y_j at nodes of a target competitive field F_2. At active F_2 nodes, instar learning [eqn (57)] strengthens the net signal sent by the active input $\mathbf{I}$. In general, codes generated by competitive learning networks are unstable, never converging to a consistent representation for certain repeated input sequences (Grossberg, 1976a; Carpenter & Grossberg, 1987). In particular, with fast learning and activation $\mathbf{y}$ distributed across all F_2 nodes, all the weights w_{ij} would converge to the same value, I_i, with each input presentation, a form of catastrophic forgetting.

The $F_0 \rightarrow F_2$ portion of the dART network, with distributed instar learning [eqn (48)] and with F_2 signals and activation as described in Section 2, constitutes a distributed competitive learning system. This network is, in fact, equivalent to the dART network with vigilance

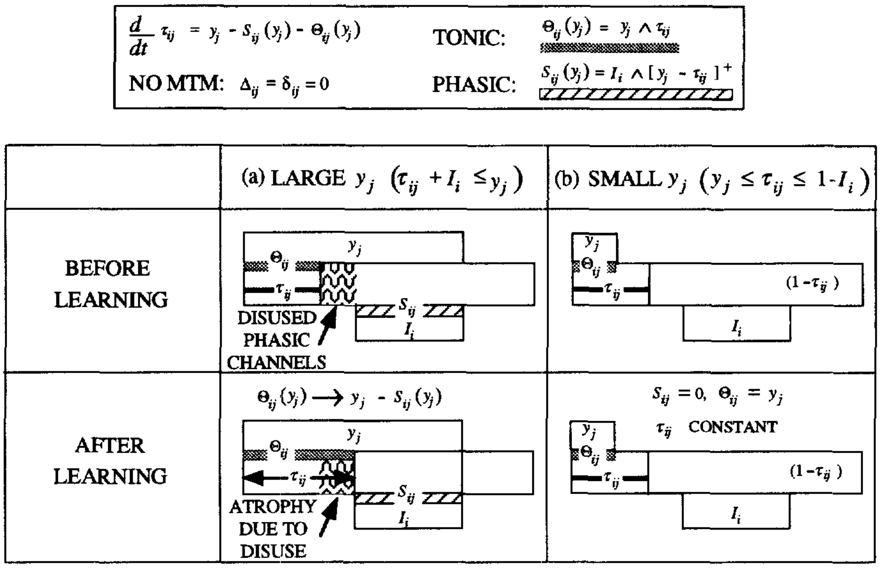
$\frac{d}{dt}\tau_{ij} = y_j - S_{ij}(y_j) - \Theta_{ij}(y_j)$
TONIC:
$\Theta_{ij}(y_j) = y_j \wedge \tau_{ij}$
NO MTM: $\Delta_{ij} = \delta_{ij} = 0$
PHASIC:
$S_{ij}(y_j) = I_i \wedge [y_j - \tau_{ij}]^+$
(a) LARGE y_j $(\tau_{ij} + I_i \le y_j)$
(b) SMALL y_j $(y_j \le \tau_{ij} \le 1 - I_i)$
BEFORE LEARNING
y_j
Θ_{ij}
τ_{ij}
S_{ij}
I_i
DISUSED PHASIC CHANNELS
$(1-\tau_{ij})$
AFTER LEARNING
$\Theta_{ij}(y_j) \to y_j - S_{ij}(y_j)$
ATROPHY DUE TO DISUSE
$S_{ij} = 0,\ \Theta_{ij} = y_j$
τ_{ij} CONSTANT

FIGURE 4. Distributed instar learning at synapse i of the j^{th} F_2 node: disused phasic channels (pattern) that are primed by y_j but not occupied by I_i revert to tonic channels; (a) a large y_j may permit the threshold τ_{ij} to increase during learning. When τ_{ij} is increasing, the tonic terms increase because then $\Theta_{ij}(y_j) = \Theta_{ij}(1) = \tau_{ij}$ while the phasic terms remain constant because then $S_{ij}(y_j) = S_{ij}(1) = I_i$; (b) a small y_j tends to leave τ_{ij} constant during learning because then $\Theta_{ij}(y_j) = y_j$ and $S_{ij}(y_j) = 0$.

$\rho \equiv 0$, which eliminates the match/reset/search cycle. The $F_0 \rightarrow F_2$ competitive network is thus a special case of a distributed ART network. The key to this distributed competitive learning design is the dynamic weight $[y_j - \tau_{ij}]^+$ that replaces the traditional multiplicative weight w_{ij}. The distributed instar learning law [eqn (48)] holds constant all thresholds τ_{ij} greater than $(y_j - I_i)$. Adaptive increase of a threshold τ_{ij} requires a combination of a relatively small starting value $\tau_{ij}(r)$, a small path input I_i, and large coding node activation y_j. Since $|\mathbf{y}| = 1$ but each $I_i \in [0,1]$, most thresholds will remain unchanged during learning. When the inequality $\tau_{ij}(r) < (y_j - I_i)$ permits adaptation, τ_{ij} rises only toward the upper limit $(y_j - I_i)$, where the dynamic weight $[y_j - \tau_{ij}]^+$ equals the input I_i. In contrast, instar learning [eqn (57)] permits adaptation for all positive y_j.

During distributed instar learning with a given input **I**, the F_2 code **y** remains constant. However, learning may alter the code that this same input will activate later, as follows. Recall that **y** is determined by the size of the $F_0 \rightarrow F_2$ signal T_j (1) at the time $t = r$ of the previous reset. By eqns (24) and (25), each y_j is an increasing function of $T_j\,(1)|_{t=r}$. During learning, the quantity $I_i \wedge [y_j - \tau_{ij}(t)]^+$ in the phasic term $S_{ij}(y_j)$ [eqn (9)] remains constant, since $[y_j - \tau_{ij}(t)]^+$ decreases only if it is greater than I_i (Figure 4). In contrast, the quantity $y_j \wedge \tau_{ij}(t)$ in the tonic term $\Theta_{ij}(y_j)$ [eqn (13)] increases whenever $\tau_{ij}(t)$ increases, since $\tau_{ij}(t)$ can increase only if $y_j > \tau_{ij}(t)$ [eqn (48)]. If **I** is presented again at a later time with no other learned changes having occurred, each term $S_{ij}\,(1) = I_i \wedge (1 - \tau_{ij})$ will be the same as it had been when **I** was previously presented and each term $\Theta_{ij}\,(1) = \tau_{ij}$ will be the same or larger. Thus by eqns (5) and (6), each increase in a threshold τ_{ij} increases the net signal T_j (1) produced by the same input **I**, all other things being equal. Since **y** is normalized, learning tends to contrast enhance the F_2 coding pattern activated by a given input: learned changes tend to occur at nodes where y_j is large, so T_j (1) becomes larger and the j^{th} node will tend to gain an advantage the next time **I** is presented.

Whereas learning can only increase the $F_0 \rightarrow F_2$ signals T_j (1) for the active input **I**, subsequent learned changes associated with different inputs could cause either an increase or a decrease in T_j (1) the next time **I** is presented. Note that τ_{ij} increases when an active F_2 node j is coding an input in which the i^{th} component is small [eqn (48)]. The computations below show that, if this happens, the next time input **I** is presented the larger threshold τ_{ij} will cause a larger signal T_j (1) where I_i is small but will cause a smaller T_j (1) where I_i is large. That is, learning has caused node j to become more responsive to the set of all inputs where the i^{th} component is small.

Suppose that the last time **I** was presented, τ_{ij} was equal to τ_{ij}^{old} but that τ_{ij} has, in the mean time, risen to τ_{ij}^{new}. Suppose that **I** is now presented again. If I_i is small $\left(I_i \le 1 - \tau_{ij}^{\text{new}}\right)$, then the phasic term S_{ij} (1) will be the same as before but the tonic term Θ_{ij} (1) will have increased from τ_{ij}^{old} to τ_{ij}^{new} [Figure 5(a)]. Thus, when I_i is small an increased threshold τ_{ij} leads to a larger signal T_j (1), by eqns (5), (6) and (12). With choice-by-difference [eqn (15)], T_j (1) increases by $(1-\alpha)\left(\tau_{ij}^{\text{new}} - \tau_{ij}^{\text{old}}\right)$. If I_i is large $\left(I_i \ge 1 - \tau_{ij}^{\text{old}}\right)$, then S_{ij} (1) will have decreased from $\left(1 - \tau_{ij}^{\text{old}}\right)$ to $\left(1 - \tau_{ij}^{\text{new}}\right)$ while Θ_{ij} (1) will have increased by the same amount, from τ_{ij}^{old} to τ_{ij}^{new} [Figure 5(c)]. Thus, eqn (6) implies that when I_i is large, an increased threshold τ_{ij} leads to a smaller signal T_j (1). With choice-by-difference, T_j (1) decreases by $\alpha\left(\tau_{ij}^{\text{new}} - \tau_{ij}^{\text{old}}\right)$. If I_i is in

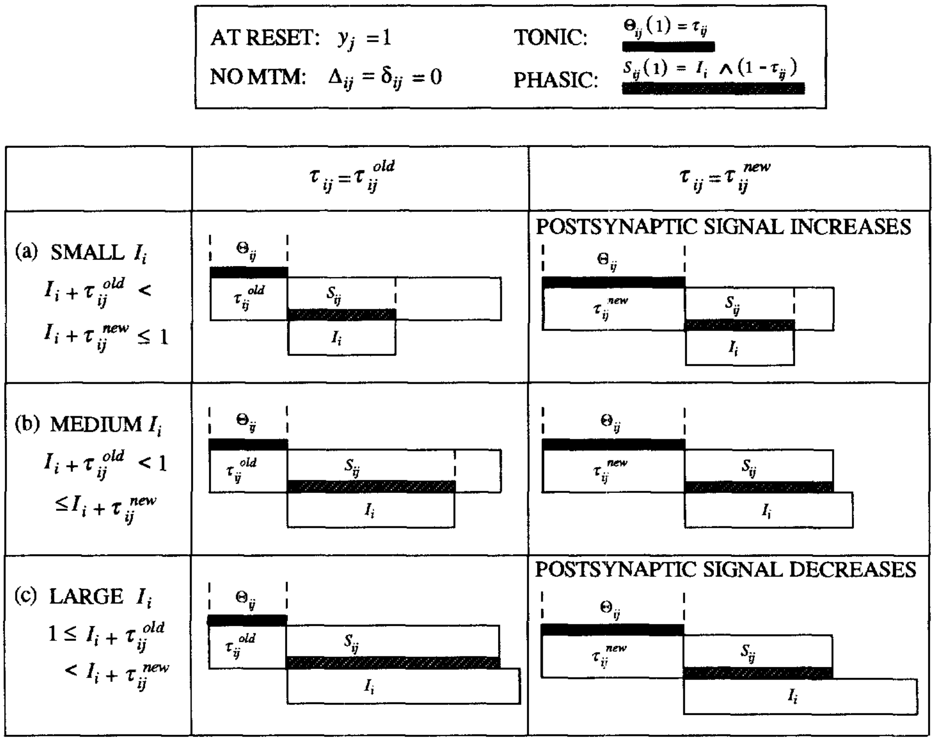
AT RESET: $y_j = 1$
NO MTM: $\Delta_{ij} = \delta_{ij} = 0$
TONIC: $\Theta_{ij}(1) = \tau_{ij}$
PHASIC: $S_{ij}(1) = I_i \wedge (1 - \tau_{ij})$
$\tau_{ij} = \tau_{ij}^{old}$
$\tau_{ij} = \tau_{ij}^{new}$
(a) SMALL I_i
$I_i + \tau_{ij}^{old} <$
$I_i + \tau_{ij}^{new} \le 1$
POSTSYNAPTIC SIGNAL INCREASES
(b) MEDIUM I_i
$I_i + \tau_{ij}^{old} < 1$
$\le I_i + \tau_{ij}^{new}$
(c) LARGE I_i
$1 \le I_i + \tau_{ij}^{old}$
$< I_i + \tau_{ij}^{new}$
POSTSYNAPTIC SIGNAL DECREASES
Θ_{ij}
S_{ij}
I_i
τ_{ij}^{old}
τ_{ij}^{new}

FIGURE 5. Effect of learned changes on coding signals: an increase in the threshold τ_{ij} between presentations of an input I may make $T_j(1)$ larger or smaller the next time I is presented, depending on the size of I_i. That is, although learning causes a monotonic change in the LTM representation at the level of receptors (τ_{ij}), this change can resemble either LTP (for a single test pulse or small I_i) or LTD (for larger I_i) at the level of the postsynaptic potential (y_j): (a) when I_i is small, a higher threshold τ_{ij} makes the tonic term Θ_{ij} larger while the phasic term S_{ij} stays the same, so $T_j(1)$ is larger; (b) when I_i is neither large nor small, a higher threshold makes Θ_{ij} larger and S_{ij} smaller, and $T_j(1)$ may be larger or smaller, depending on the signal rule that defines it; (c) when I_i is large, a higher threshold increases Θ_{ij} and decreases S_{ij} by equal amounts, making $T_j(1)$ smaller.

between $\left(1 - \tau_{ij}^{\text{new}} \le I_i < 1 - \tau_{ij}^{\text{old}}\right)$, an increased threshold τ_{ij} may lead either to a smaller or a larger signal T_j (1), depending on the function $g_j(S_j,\Theta_j)$ that defines T_j [eqn (5)]. The phasic term S_{ij} (1) will have decreased from I_i to $\left(1 - \tau_{ij}^{\text{new}}\right)$ while the tonic term Θ_{ij} (1) will have increased from τ_{ij}^{old} to τ_{ij}^{new} [Figure 5(b)]. With choice-by-difference, the change in T_j (1) is:

$$\Delta T_j(1) = \left(1 - \tau_{ij}^{\text{new}} - I_i\right) + (1-\alpha)\left(\tau_{ij}^{\text{new}} - \tau_{ij}^{\text{old}}\right) \qquad (60)$$

$$= \left(1 - \tau_{ij}^{\text{old}} - I_i\right) - \alpha\left(\tau_{ij}^{\text{new}} - \tau_{ij}^{\text{old}}\right).$$

Thus, the increased threshold τ_{ij} leads to a larger signal T_j (1) only if the choice parameter α is small enough; that is if:

$$(1-\alpha)\tau_{ij}^{\text{old}} + \alpha\tau_{ij}^{\text{new}} < (1 - I_i). \qquad (61)$$

5. A DISTRIBUTED ART ALGORITHM

The algorithm below summarizes distributed ART [Figure 1(b)] computations with inputs $\mathbf{I}^{(1)}$, $\mathbf{I}^{(2)}$,..., $\mathbf{I}^{(n)}$,... presented for equal time intervals. An algorithm to approximate dART dynamics for a continuously varying input $\mathbf{I}(t)$ would set $\mathbf{I}^{(n)} = \mathbf{I}(n\Delta t)$, with the time step Δt and the learning rate parameter β small. Other dART variations are implemented with appropriate substitutions.

(1) Variables: $i = 1...M, j = 1...N$

STM	MTM	LTM	$F_0 \to F_2$ signal	$F_2 \to F_1$ signal
I_i – F_0 (input)	Δ_{ij} – Phasic	τ_{ij} – $F_0 \to F_2$	S_j – Phasic	σ_i – Total
x_i – F_1 (matching)	δ_{ij} – Tonic	τ_{ji} – $F_2 \to F_1$	Θ_j – Tonic	
y_j – F_2 (coding)			T_j – Total	

(2) Signal rule: define the $F_0 \to F_2$ signal function $T_j = g_j(S_j, \Theta_j)$, where $g_j(0,0) = 0$ and $\partial g_j / \partial S_j > \partial g_j / \partial \Theta_j > 0$ for $S_j > 0$ and $\Theta_j > 0$.

For example, $T_j = S_j + (1 - \alpha)\Theta_j$ with $\alpha \in (0,1)$ (choice-by-difference) or $T_j = S_j / (\alpha + My_j - \Theta_j)$ with $\alpha > 0$ (Weber law).

(3) CAM rule: define the F_2 steady-state function $y_j = f_j(T_1...T_N)$, where $\partial f_j / \partial T_j \ge 0$.

For example, for a power $p > 0$ (power law)

$$y_j = \left\{ \begin{array}{ll} \dfrac{(T_j)^p}{\sum\limits_{\lambda \in \Lambda} (T_\lambda)^p} & \text{if } j \in \Lambda \\ 0 & \text{if } j \notin \Lambda \end{array} \right\}$$

where

$$\Lambda = \{j : \; T_j \ge \overline{T}\} \text{ with } \overline{T} = \frac{1}{N}\sum_{j=1}^{N} T_j$$

(above average—T_j); or Λ = the set of Q indices j where T_j is maximal (Q-max).

(4) Parameters:

Number of input components, $i = 1 \ldots M$
Number of coding nodes, $j = 1 \ldots N$
Signal rule, e.g., $\alpha \in (0,1)$ (choice-by-difference) or $\alpha > 0$ (Weber law)
CAM rule, e.g., p (power law) and Q (Q-max), with $p \rightarrow \infty$ or $Q = 1$ for choice
Learning rate, $\beta \in [0,1]$, with $\beta = 1$ for fast learning
Vigilance, $\rho \in [0,1]$
A set of small, positive, random numbers, for initial τ_{ij} values, $\eta_{ij} = 0^+$

(5) First iteration: $n = 1$

MTM depletion	$\Delta_{ij} = \delta_{ij} = 0$
$F_0 \rightarrow F_2$ threshold	$\tau_{ij} = \eta_{ij}$
$F_2 \rightarrow F_1$ threshold	$\tau_{ji} = 0$
Input	$I_i = I_i^{(1)}$

(6) Reset: new STM steady-state at F_2 and F_1

$F_0 \rightarrow F_2$ signal Phasic	$S_j = \sum_{i=1}^{M} \left[I_i \wedge (1 - \tau_{ij}) - \Delta_{ij}\right]^+$
Tonic	$\Theta_j = \sum_{i=1}^{M} \left[\tau_{ij} - \delta_{ij}\right]^+$
Total	$T_j = g_j(S_j, \Theta_j)$ [(2) Signal rule]
F_2 activation	$y_j = f_j(T_1 \ldots T_N)$ [(3) CAM rule]
$F_2 \rightarrow F_1$ signal	$\sigma_i = \sum_{j=1}^{N} \left[y_j - \tau_{ji}\right]^+$
F_1 activation	$x_i = I_i \wedge \sigma_i$

(7) MTM depletion: F_2 sites refractory on the time scale of search

Phasic	$\Delta_{ij}^{\text{old}} = \Delta_{ij}$ $\Delta_{ij} = \Delta_{ij}^{\text{old}} \vee \left(I_i \wedge \left[y_j - \tau_{ij}\right]^+\right)$
Tonic	$\delta_{ij}^{\text{old}} = \delta_{ij}$ $\delta_{ij} = \delta_{ij}^{\text{old}} \vee (y_j \wedge \tau_{ij})$

(8) Reset or resonance: check the F_1 matching criterion

If $\sum_{i=1}^{M} x_i < \rho \sum_{i=1}^{M} I_i$, go to (6) Reset

If $\sum_{i=1}^{M} x_i \geq \rho \sum_{i=1}^{M} I_i$, go to (9) Resonance

(9) Resonance: new LTM thresholds and MTM recovery on the time scale of learning

Old values	$\tau_{ij}^{\text{old}} = \tau_{ij}, \quad \tau_{ji}^{\text{old}} = \tau_{ji}, \quad \sigma_i^{\text{old}} = \sigma_i$
Increase $F_0 \rightarrow F_2$ threshold	$\tau_{ij} = \tau_{ij}^{\text{old}} + \beta\left[y_j - \tau_{ij}^{\text{old}} - I_i\right]^+$
Increase $F_2 \rightarrow F_1$ threshold	$\tau_{ji} = \tau_{ji}^{\text{old}} + \beta \dfrac{\left[\sigma_i^{\text{old}} - I_i\right]^+}{\sigma_i^{\text{old}}} \left[y_j - \tau_{ji}^{\text{old}}\right]^+$
Decrease $F_2 \rightarrow F_1$ signal	$\sigma_i = \sigma_i^{\text{old}} - \beta\left[\sigma_i^{\text{old}} - I_i\right]^+$
MTM recovery	$\Delta_{ij} = \delta_{ij} = 0$

(10) Next iteration: increase n by 1

New input	$I_i = I_i^{(n)}$
New F_1 activation	$x_i = I_i \wedge \sigma_i$
Go to (6) Reset	

6. DISTRIBUTED ART GEOMETRY

A geometric interpretation of fuzzy ART represents categories as boxes in input space that expand during learning (Carpenter et al., 1991b). A generalized version of this geometric representation illustrates dART dynamics, as follows.

6.1. Complement Coding

In fuzzy ART, input normalization prevents a type of category proliferation that could otherwise occur when weights erode. Complement coding doubles the dimension of an input vector $\mathbf{a} \equiv (a_1 \ldots a_M)$ by concatenating $\mathbf{a}$ and its complement $\mathbf{a}^c$. The input to a fuzzy ART network is then a $2M$-dimensional vector:

$$\mathbf{I} = \mathbf{A} \equiv \left(\mathbf{a}, \mathbf{a}^c\right), \tag{62}$$

where

$$\left(\mathbf{a}^c\right)_i \equiv \left(1 - a_i\right). \tag{63}$$

Complement coded inputs are normalized because

$$|\mathbf{A}| = \left|\left(\mathbf{a}, \mathbf{a}^c\right)\right| = \sum_{i=1}^{M} a_i + \sum_{i=1}^{M} \left(1 - a_i\right) = M. \tag{64}$$

If $\mathbf{a}$ represents input features, then complement coding allows a learned category representation to encode the degree to which each feature is consistently absent as well as the degree to which it is consistently present when that category is active. Because of its computational advantages, complement coding is used in nearly all fuzzy ART and fuzzy ARTMAP applications. Similar advantages can be expected for dART and dARTMAP applications. Except for changing the number of components of $\mathbf{I}$, $\mathbf{x}$, and the corresponding LTM vectors from M to $2M$, the description of network dynamics is unchanged since complement coding is only a preprocessing step. A dART algorithm with complement coding can be embedded in an ARTMAP network to form the basis of a dARTMAP algorithm (Section 8).

6.2. Fuzzy ART Category Boxes

A geometric interpretation of fuzzy ART associates with each weight vector $\mathbf{w}_j \equiv (w_{1j} \ldots w_{2M,j})$ a box R_j in M-dimensional space. In the i^{th} dimension ($i = 1 \ldots M$), the side of the j^{th} box is defined by the interval $\left[w_{ij}, w_{i+M,j}^c\right]$. That is, R_j is the set of points $\mathbf{q}$ for which:

$$w_{ij} \leq q_i \leq \left(1 - w_{i+M,j}\right). \tag{65}$$

The size of R_j is defined as the sum of these intervals:

$$\begin{aligned} |R_j| &= \sum_{i=1}^{M} \left(\left(1 - w_{i+M,j}\right) - w_{ij}\right) \\ &= M - \sum_{i=1}^{2M} w_{ij} = M - |\mathbf{w}_j|. \end{aligned} \tag{66}$$

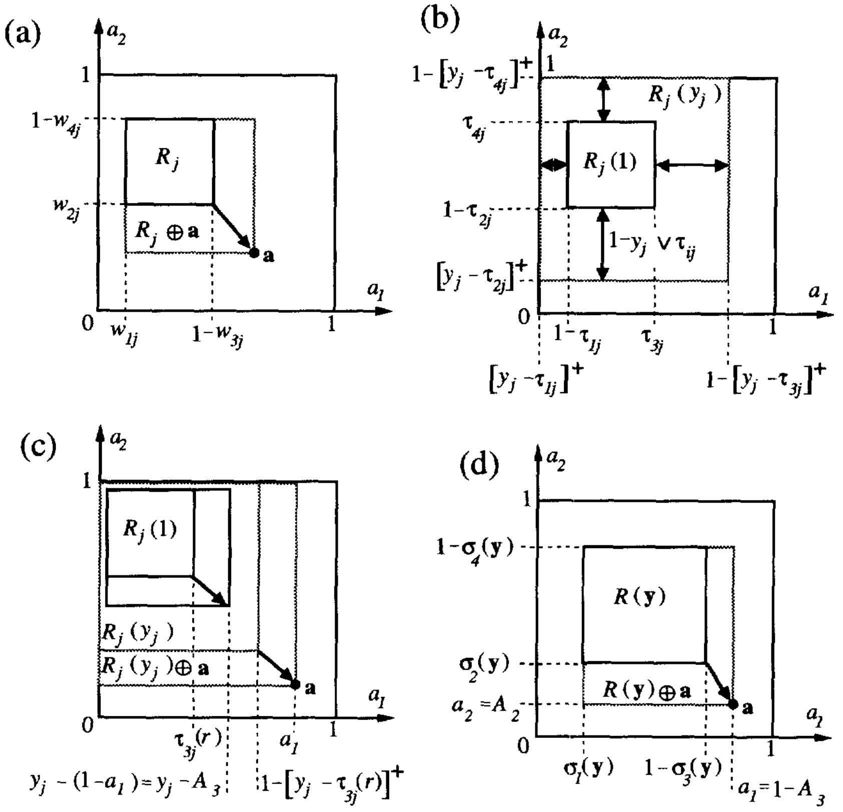

FIGURE 6. ART and dART geometry: (a) the fuzzy ART category box R_j provides a geometric representation of each weight vector $\mathbf{w}_j$. Since bottom–up and top–down weight vectors are equal, category boxes can represent the dynamics of choice, search, and learning at F_1 and F_2. During learning, the chosen box R_j expands towards $R_j \oplus \mathbf{a}$. Before this can occur, however, the search process resets node j and sends T_j to 0 if the size of the expanded box $R_j \oplus \mathbf{a}$ would be greater than $M(1-\rho)$; (b) distributed ART replaces the bottom–up fuzzy ART weights w_{ij} with a family of dynamic weights $[y_j - \tau ij]^+$ and replaces the category box R_j with a corresponding nested family of coding boxes $R_j(y_j)$. The dART box $R_j(1)$ corresponds to the fuzzy ART box R_j; (c) during distributed instar learning, with activity y_j at the j^{th} node, the box $R_j(y_j)$expands toward $R_j(y_j) \oplus \mathbf{a}$ ($j = 1 \ldots N$) as some adaptive thresholds τij increase ($i = 1 \ldots 2M$). Since $R_j(0)$ fills the square, no thresholds τij change when $y_j = 0$. The boxes $R_j(1)$ that will determine the next code $\mathbf{y}$ expand as much as the larger boxes $R_j(y_j)$. However, $R_j(1)$ will reach $\mathbf{a}$ only if $y_j = 1$ or if $\mathbf{a}$ was already contained in $R_j(1)$ at the time of the previous reset; (d) when the code $\mathbf{y}$ is active, a matching box $R(\mathbf{y})$ represents the $F_2 \rightarrow F_1$ inputs $\sigma_i(\mathbf{y})$. With choice at F_2, $\sigma_i(\mathbf{y}) = (1 - \tau Ji) \equiv w_{Ji}$, and $R(\mathbf{y})$ corresponds to the fuzzy ART box R_J. The code $\mathbf{y}$ will be reset if $R(\mathbf{y}) \oplus \mathbf{a}$ is greater than $M(1-\rho)$. If $|R(\mathbf{y}) \oplus \mathbf{a}| \leq M(1-\rho)$, $R(\mathbf{y})$expands toward $R(\mathbf{y}) \oplus \mathbf{a}$ during distributed outstar learning, as thresholds τji increase and $\sigma i(\mathbf{y})$ converges toward $\sigma i(\mathbf{y}) \wedge A_i$.

When $M = 2$,

$$\mathbf{A} = (\mathbf{a}, \mathbf{a}^c) \equiv (a_1, a_2, a_1^c, a_2^c) \tag{67}$$

and category boxes are rectangles in the plane [Figure 6(a)]. Note that, formally, the interval would be "reversed" if $w_{ij} > (1 - w_{i+M,j})$. Initially, all $w_{ij} = 1$ and $|\mathbf{w}_j| = 2M$, so initially $|R_j| = -M$. During learning, R_j may grow toward a maximum size M as weights shrink. Because top–down weights w_{ji} equal bottom–up weights w_{ij} in fuzzy ART, R_j can represent both.

When a fuzzy ART F_2 node j is chosen, $\sigma(\mathbf{y}) = \mathbf{w}_j$ and the matched F_1 pattern $\mathbf{x} = \mathbf{I} \wedge \mathbf{w}_j$ must satisfy the vigilance criterion [eqn (30)] for j to remain active [Figure 1(a)]. This is equivalent to requiring that, for category j to remain active during a learning interval,

$$|R_j \oplus \mathbf{a}| \leq M(1-\rho) \tag{68}$$

where $R_j \oplus \mathbf{a}$ is the smallest box containing both R_j and $\mathbf{a}$. When the j^{th} F_2 node does remain active, instar learning [eqn (57)] implies that w_{ij} may decrease toward $a_i = A_i$ and $w_{i+M,j}$ may decrease toward $a_i^c = A_{i+M}$. As weights shrink, the size of the interval $[w_{ij}, w_{i+M,j}^c]$ expands. A wide interval signals that the i^{th} feature is uninformative with respect to the j^{th} category: since both weights w_{ij} and $w_{i+M,j}$ are then small, the corresponding feature has been neither consistently present nor consistently absent when the j^{th} F_2 node has been active. When node j remains active during a fast learning interval, box R_j expands to $R_j \oplus \mathbf{a}$. Thus, with category choice and fast learning, R_j is the smallest box that contains all the training set inputs $\mathbf{a}$ coded by category j.

6.3. Distributed ART Coding Boxes and Matching Boxes

A geometric representation of distributed ART substitutes dART dynamic weights $[y_j - \tau_{ij}]^+$ for the fuzzy ART weights w_{ij}. For each $j = 1 \ldots N$, where fuzzy ART weights define a single category box R_j, dART dynamic weights define a family of *coding boxes* R_j (y_j), one for each $y_j \in [0,1]$. Fuzzy ART boxes R_j can represent top–down matching as well as bottom–up category activation since only one F_2 node at a time is active and $w_{ji} = w_{ij}$. In dART, however, the $F_2 \rightarrow F_1$ input vector $\sigma(\mathbf{y})$ [eqn (28)] may depend on activities of all F_2 nodes. Top–down dynamic weights $[y_j - \tau_{ji}]^+$ therefore define a *matching box* $R(\mathbf{y})$ for each F_2 activity vector $\mathbf{y}$.

A distributed ART coding box $R_j(y_j)$ depends on the F_2 activity level y_j as well as on the $F_0 \rightarrow F_2$ thresholds τ_{ij} $(i = 1\ldots 2M)$. For each $y_j \in [0,1]$, $R_j(y_j)$ is the set of points $\mathbf{q}$ for which:

$$[y_j - \tau_{ij}]^+ \le q_i \le \left(1 - [y_j - \tau_{i+M,j}]^+\right) \qquad (69)$$

[Figure 6(b)]. With $w_{ij} \equiv 1 - \tau_{ij}$ and $y_j = 1$, the dART coding box $R_j(y_j)$ is the same as the fuzzy ART category box R_j. As y_j decreases from 1 to 0, the box $R_j(y_j)$ grows, filling the entire unit box when y_j is smaller than all the thresholds τ_{ij} $(i = 1\ldots 2M)$, i.e., when all the dynamic weights $[y_j - \tau_{ji}]^+$ equal 0. Ignoring MTM aftereffects (i.e., with $\Delta_{ij} = \delta_{ij} = 0$), the size of $R_j(y_j)$ is:

$$|R_j(y_j)| = \sum_{i=1}^{M}\left(\left(1 - [y_j - \tau_{i+M,j}]^+\right) - [y_j - \tau_{ij}]^+\right) \qquad (70)$$

$$= M - \sum_{i=1}^{2M}[y_j - \tau_{ij}]^+ = M - \sum_{i=1}^{2M}(y_j - y_j \wedge \tau_{ij})$$

$$= M - 2My_j + \sum_{i=1}^{2M} y_j \wedge \tau_{ij} = M(1 - 2y_j) + \Theta_j(y_j).$$

Thus $R_j(y_j)$ represents the tonic component $\Theta_j(y_j)$ [eqns (12) and (13)] of the $F_0 \rightarrow F_2$ signal $T_j(y_j)$ [eqn (5)]. The expanded box $R_j(y_j) \oplus \mathbf{a}$ represents the phasic component $S_j(y_j)$ [eqns (8) and (9)], as follows.

For a given input $\mathbf{a} \equiv (a_1\ldots a_M)$, $R_j(y_j) \oplus \mathbf{a}$ is the set of points $\mathbf{q}$ where:

$$\left\{[y_j - \tau_{ij}]^+ \wedge a_i\right\} \le q_i \le \left\{\left(1 - [y_j - \tau_{i+M,j}]^+\right) \vee a_i\right\} \qquad (71)$$

[Figure 6(c)]. For $i = 1\ldots M$,

$$[y_j - \tau_{ij}]^+ \wedge a_i = [y_j - \tau_{ij}]^+ \wedge A_i \qquad (72)$$

and:

$$\left(1 - [y_j - \tau_{i+M,j}]^+\right) \vee a_i = \left(1 - [y_j - \tau_{i+M,j}]^+\right) \vee \left(1 - (1 - a_i)\right) \qquad (73)$$

$$= 1 - [y_j - \tau_{i+M,j}]^+ \wedge (1 - a_i)$$

$$= 1 - [y_j - \tau_{i+M,j}]^+ \wedge A_{i+M}.$$

Thus

$$|R_j(y_j) \oplus \mathbf{a}| = \sum_{i=1}^{M}\left(\left\{\left(1 - [y_j - \tau_{i+M,j}]^+\right) \vee a_i\right\} - \left\{[y_j - \tau_{ij}]^+ \wedge a_i\right\}\right) \qquad (74)$$

$$= \sum_{i=1}^{M}\left(1 - [y_j - \tau_{i+M,j}]^+ \wedge A_{i+M} - [y_j - \tau_{ij}]^+ \wedge A_i\right)$$

$$= M - \sum_{i=1}^{2M}[y_j - \tau_{ij}]^+ \wedge A_i = M - S_j(y_j).$$

Therefore, the expanded box $R_j(y_j) \oplus \mathbf{a}$ represents the phasic component $S_j(y_j)$ of the $F_0 \rightarrow F_2$ signal $T_j(y_j)$.

The boxes $R_j(y_j)$ and $R_j(y_j) \oplus \mathbf{a}$ also provide a geometric representation of the distributed choice-by-difference signal rule. Defining the distance $d(R_j,\mathbf{a})$ from R_j to $\mathbf{a}$ by:

$$d(R_j, \mathbf{a}) \equiv |R_j \oplus \mathbf{a}| - |R_j|, \qquad (75)$$

eqns (70) and (74) imply that the choice-by-difference signal function [eqn (15)] can be written as:

$$T_j(y_j) = S_j(y_j) + (1 - \alpha)\Theta_j(y_j) \qquad (76)$$

$$= \left(M - |R_j(y_j) \oplus \mathbf{a}|\right) + (1 - \alpha)\left(|R_j(y_j)| - M(1 - 2y_j)\right)$$

$$= M\left(1 - (1 - \alpha)(1 - 2y_j)\right) - d\left(R_j(y_j), \mathbf{a}\right) - \alpha|R_j(y_j)|.$$

Recall that the values $y_1\ldots y_N$ will assume following a reset are determined by $T_1(1)\ldots T_N(1)$ [eqn (24)]. By eqn (76),

$$T_j(1) = \left(M - |R_j(1) \oplus \mathbf{a}|\right) + (1 - \alpha)\left(|R_j(1)| + M\right) \qquad (77)$$

$$= M(2 - \alpha) - d\left(R_j(1), \mathbf{a}\right) - \alpha|R_j(1)|.$$

Geometric interpretation of distributed choice-by-difference thus shows that, except for MTM aftereffects during search, an input $\mathbf{a}$ will most strongly activate an F_2 node j when $\mathbf{a}$ is in or near $R_j(1)$ and when $R_j(1)$ is small. The relative importance of distance vs size depends on the choice parameter α. When α is close to 0, a maximal $T_j(1)$ is one that minimizes the distance from $R_j(1)$ to $\mathbf{a}$, with the size of $R_j(1)$ used only to break ties if $\mathbf{a}$ is contained in more than one coding box. When α is close to 1, a maximal $T_j(1)$ is one that minimizes the size of the expanded box $R_j(1) \oplus \mathbf{a}$.

During distributed instar learning, while $\mathbf{a}$ and $\mathbf{y}$ remain constant, $R_j(y_j)$ expands toward $R_j(y_j) \oplus \mathbf{a}$ [Figure 6(c)] as the tonic terms $y_j \wedge \tau_{ij}$ in $\Theta_j(y_j)$ grow and the phasic terms $A_i \wedge [y_j - \tau_{ij}]^+$ in $S_j(y_j)$ remains constant. No threshold τ_{ij} will change during learning if $\mathbf{a}$ is contained in $R_j(y_j)$, even if $\mathbf{a}$ is not contained in $R_j(1)$. With fast learning following a reset at time $t = r$, thresholds grow from $\tau_{ij}^{\text{old}} \equiv \tau_{ij}(r)$ to τ_{ij}^{new} and:

$$\sum_{i=1}^{2M} |\Delta\tau_{ij}| \equiv \sum_{i=1}^{2M} \left(\tau_{ij}^{\text{new}} - \tau_{ij}^{\text{old}}\right) = \sum_{i=1}^{2M} \left[y_j - A_i - \tau_{ij}^{\text{old}}\right]^+ \tag{78}$$

$$= \sum_{i=1}^{2M} \left(\left[y_j - \tau_{ij}^{\text{old}}\right]^+ - \left[y_j - \tau_{ij}^{\text{old}}\right]^+ \wedge A_i \right)$$

$$= \sum_{i=1}^{2M} \left[y_j - \tau_{ij}^{\text{old}}\right]^+ - \sum_{i=1}^{2M} \left[y_j - \tau_{ij}^{\text{old}}\right]^+ \wedge A_i$$

$$= \left(M - |R_j^{\text{old}}(y_j)|\right) - \left(M - |(R_j^{\text{old}}(y_j) \oplus \mathbf{a}|\right)$$

$$= d\left(R_j(y_j), \mathbf{a}\right)|_{t=r}.$$

That is, the total threshold change at the j^{th} F_2 node equals the distance from $R_j(y_j)$ to $\mathbf{a}$ at time $t = r$, the start of the learning interval. Thus, setting $\alpha = 0^+$, which favors F_2 nodes for which $R_j(1)$ is closest to $\mathbf{a}$ in eqn (77), also favors nodes that will minimize the total $F_0 \rightarrow F_2$ threshold change during learning. In fuzzy ART, the parameter limit where α is close to 0 was called the *conservative limit*, since category choice then favors weight conservation wherever possible.

Once a dART code $\mathbf{y}$ becomes active, the signal $\sigma_i(\mathbf{y})$ from F_2 to the i^{th} F_1 node equals the sum of the top–down dynamic weights $[y_j - \tau_{ji}]^+$. The signals $\sigma_i(\mathbf{y})$ define a matching box $R(\mathbf{y})$ as the set of points $\mathbf{q}$ where:

$$\sigma_i(\mathbf{y}) \leq q_i \leq 1 - \sigma_{i+M}(\mathbf{y}) \tag{79}$$

for $i = 1 \ldots M$ [Figure 6(d)]. The expanded box $R(\mathbf{y}) \oplus \mathbf{a}$ is the set of points $\mathbf{q}$ where:

$$\sigma_i(\mathbf{y}) \wedge a_i \leq q_i \leq \left(1 - \sigma_{i+M}(\mathbf{y})\right) \vee a_i. \tag{80}$$

As in eqns (72) and (73),

$$\sigma_i(\mathbf{y}) \wedge a_i = \sigma_i(\mathbf{y}) \wedge A_i \tag{81}$$

and:

$$\left(1 - \sigma_{i+M}(\mathbf{y})\right) \vee a_i = 1 - \sigma_{i+M}(\mathbf{y}) \wedge A_{i+M} \tag{82}$$

for $i = 1 \ldots M$. Thus, by eqns (29), (64) and (80),

$$|R(\mathbf{y}) \oplus \mathbf{a}| = \sum_{i=1}^{M} \left(\left\{ \left(1 - \sigma_{i+M}(\mathbf{y})\right) \vee a_i \right\} - \left\{ \sigma_i(\mathbf{y}) \wedge a_i \right\} \right) \tag{83}$$

$$= M - \sum_{i=1}^{2M} \sigma_i(\mathbf{y}) \wedge A_i = |\mathbf{A}| - |\sigma(\mathbf{y}) \wedge \mathbf{A}| = |\mathbf{A}| - |\mathbf{x}|.$$

Therefore, the active dART code $\mathbf{y}$ meets the matching criterion [eqn (31)] when:

$$|R(\mathbf{y}) \oplus \mathbf{a}| \leq M(1 - \rho), \tag{84}$$

as in fuzzy ART [eqn (68)]. Geometrically, resonance requires that the expanded box $R(\mathbf{y}) \oplus \mathbf{a}$ not be too big, by eqn (84). When the matching criterion is met, $R(\mathbf{y})$ expands toward $R(\mathbf{y}) \oplus \mathbf{a}$ during learning. No top–down learned changes occur if $\mathbf{a}$ is already contained in $R(\mathbf{y})$.

7. DISTRIBUTED ART COMPUTATION

The dART algorithm (Section 5) summarizes a general solution to the distributed ART system of equations. Once a specific network and parameters are selected for a particular application, computational analysis is usually required to trace network coding in response to a given input sequence. When network dimensions are small, as in the examples below, explicit system solutions are simple enough to permit direct calculation, without use of a computer. Each example uses a choice-by-difference signal rule, a power law CAM rule, fast learning, and two or three coding nodes at F_2 to illustrate dART activation, search, and learning. As in Figure 6, two-dimensional inputs are complement-coded, so $\mathbf{a} = (a_1, a_2)$ and $\mathbf{I} = \mathbf{A} = (\mathbf{a}, \mathbf{a}^c)$.

7.1. dART Learning

Figure 7 illustrates distributed ART learning in a system with dimensions $M = N = 2$ and input $\mathbf{a} = (0.7, 0.8)$. The index set $\Lambda = \{1,2\} = \{1 \ldots N\}$ in the power law CAM rule, so $y_j = T_j^p(1) / \left(T_1^p(1) + T_2^p(1)\right)$ for $j = 1, 2$. For $j = 1$, initial threshold values $(\tau_{1j} \ldots \tau_{4j}) = (0.9, 0.3, 0.4, 0.9)$ are represented by the coding box $R_1(1)$, with $d(R_1(1), \mathbf{a}) = 0.3$ and $|R_1(1)| = 0.5$. For $j = 2$, initial threshold values $(\tau_{1j} \ldots \tau_{4j}) = (0.9, 0.9, 0.9, 0.2)$ are represented by the coding box $R_2(1)$, with $d(R_2(1), \mathbf{a}) = 0.6$ and $|R_2(1)| = 0.9$. By eqn (77),

$$T_j(1) = 2(2 - \alpha) - d\left(R_j(1), \mathbf{a}\right) - \alpha |R_j(1)|. \tag{85}$$

Thus when $\alpha = 0.2$, $(T_1(1), T_2(1)) = (3.20, 2.82)$ and, when $p = 1$, $(y_1, y_2) = (0.532, 0.468)$ [Figure 7(a)]. Then $\mathbf{a} \in R_1(y_1)$ but $\mathbf{a} \notin R_2(y_2)$: $d(R_1(y_1), \mathbf{a}) = 0$ and $d(R_2(y_2), \mathbf{a}) = 0.068$. During learning, $R_2(y_2)$ expands to include $\mathbf{a}$ as τ_{42} increases from 0.2 to 0.268. If $\mathbf{a}$ is repeatedly presented and no other learned changes take place, τ_{42} will continue to increase toward 0.274, the point where $(y_1, y_2) = (0.526, 0.474)$, where $R_2(y_2)$ would just include $\mathbf{a}$.

If the power p increases to 5, with the network otherwise the same, $(y_1, y_2) = (0.653, 0.347)$ [Figure 7(b)]. Compared to the case where $p = 1$, the higher power stores a more contrast-enhanced representation of the signal $(T_1(1), T_2(1))$ in the CAM system at F_2. In this case $\mathbf{a} \in R_1(y_1)$ and $\mathbf{a} \in R_2(y_2)$, so no changes occur during learning.

If the choice parameter α increases to 0.8 but the network is otherwise the same as in Figure 7(b), the signal $(T_1(1), T_2(1)) = (1.70, 1.08)$. Then $\mathbf{y}$ is further contrast-enhanced, with $(y_1, y_2) = (0.906, 0.094)$ [Figure 7(c)]. In this case, $\mathbf{a} \in R_2(y_2)$ but $\mathbf{a} \notin R_1(y_1)$: $d(R_2(y_2), \mathbf{a}) = 0$ and $d(R_1(y_1), \mathbf{a}) = 0.206$. During learning, $R_1(y_1)$ expands to include $\mathbf{a}$ as τ_{31} increases from 0.4 to 0.606.

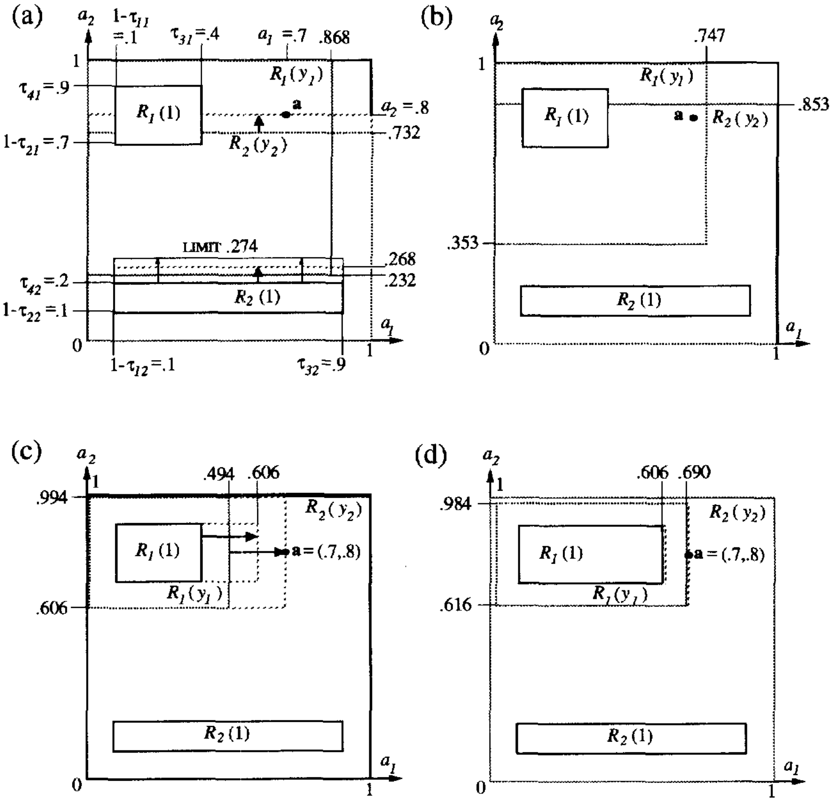

FIGURE 7. Distributed ART activation and learning in response to an input a = (0.7, 0.8)), with complement coding, a power law CAM rule, a choice-by-difference signal rule, and two coding nodes ($N = 2$): (a) when $p = 1$ and $\alpha = 0.2$, $(y_1,y_2) = (0.532, 0.468)$. During learning, $R_2(y_2)$expands to include a as τ_{42} increases from 0.2 to 0.268. If a is repeatedly presented and no other learned changes take place, τ_{42} will continue to increase toward 0.274, the point where $R_2(y_2)$ would just include a; (b) when $p = 5$ and $\alpha = 0.2$, $(y_1,y_2) = (0.653, 0.347)$ and no changes occur during learning; (c) when $p = 5$ and $\alpha = 0.8$, $(y_1,y_2) = (0.906, 0.094)$. During learning, $R_1(y_1)$expands to include a as τ_{31} increases from 0.4 to 0.606; (d) if a is presented again, $(y_1,y_2) = (0.916, 0.084)$. If a is repeatedly presented and no other learned changes take place, τ_{31} will continue to increase toward 0.616.

If **a** is presented again later with no other learned changes having taken place in the mean time (and no MTM distortion), $(y_1,y_2) = (0.916,0.084)$ [Figure 7(d)]. Learning has thus contrast enhanced the code **y**. If **a** is repeatedly presented and no other learned changes take place, τ_{31} will continue to increase toward 0.616, the point where $(y_1,y_2) = (0.916,0.084)$, where $R_1(y_1)$ would just include **a**.

Table 1 shows steady-state **y** values of the system described above (Figure 7) as the power p increases from 1 to 5 and as the choice parameter α increases from 0.01 to 0.99. During learning, $\tau_{ij} = \tau_{31}$ increases for $y_1 > 0.7$, when $\mathbf{a} \notin R_1(y_1)$, since $d(R_1(1),\mathbf{a}) = 0.3$; and $\tau_{ij} = \tau_{42}$ increases for $y_2 > 0.4$, when $\mathbf{a} \notin R_2(y_2)$, since $d(R_2(1),\mathbf{a}) = 0.6$ (boldface values of y_j). In all other cases, no changes occur during learning. If the same input **a** is presented again and no other learned changes have meanwhile occurred, a larger $\tau_{ij} = \tau_{31}$ value implies a larger rectangle $R_1(1)$, a smaller distance $d(R_1(1),\mathbf{a})$, and larger $T_1(1)$ and y_1 values. The code (y_1,y_2) is thus contrast-enhanced by the learning process. On the other hand, a larger $\tau_{ij} = \tau_{42}$ value, which implies a larger rectangle $R_2(1)$, a smaller distance $d(R_2(1),\mathbf{a})$, and larger $T_2(1)$ and y_2 values, would make the code (y_1,y_2) more uniform.

TABLE 1
Distributed ART activation and learning

	CBD Signal $(T_1(1), T_2(1))$	$p = 1$ (y_1,y_2)	$p = 2$ (y_1,y_2)	$p = 5$ (y_1,y_2)	LEARNING →	$p = 5$ (y_1,y_2)
$\alpha = 0.01$	(3.67,3.37)	(0.521,**0.479**)	(0.543,**0.457**)	(0.605,0.395)		(0.605,0.395)
$\alpha = 0.20$	(3.20,2.82)	(0.532,**0.468**)	(0.563,**0.437**)	(0.653,0.347)		(0.653,0.347)
$\alpha = 0.50$	(2.45,1.95)	(0.557,**0.443**)	(0.612,0.388)	(**0.758**,0.242)		(**0.769**,0.231)
$\alpha = 0.80$	(1.70,1.08)	(0.612,0.388)	(**0.713**,0.287)	(**0.906**,0.094)		(**0.916**,0.084)
$\alpha = 0.99$	(1.23,0.53)	(0.699,0.301)	(**0.843**,0.157)	(**0.985**,0.015)		(**0.985**,0.015)

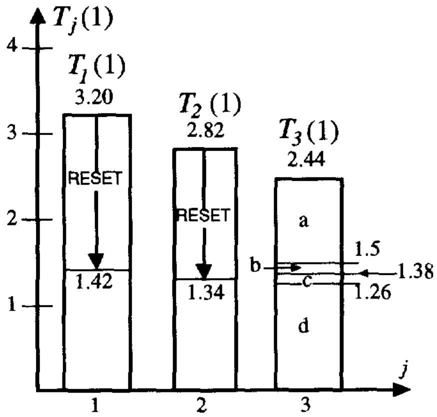

FIGURE 8. Distributed ART search in response to an input a = (0.7, 0.8)), with complement coding, a power law CAM rule ($p = 1$) for above-average $T_j(1)$, a choice-by-difference signal rule (α = 0.2), and $N = 3$. Initially, $T_1(1) = 3.20$ and $T_2(1) = 2.82$. When $T_3(1) \leq 2.44$, $T_3(1) < \bar{T} \leq T_2(1) < T_1(1)$, so $y_1 = 0.532$ and $y_2 = 0.468$, as in Figure 7(a); and $y_3 = 0$. For this code y, $T_1(y_1) = 1.78$, $T_2(y_2) = 1.48$, and $T_3(y_3) = 0$. A reset would therefore leave $T_3(1)$ unchanged but would reduce $T_1(1)$ to 1.42 and $T_2(1)$ to 1.34. The next code y then depends on the size of $T_3(1)$ [Table 2].

7.2. dART Search

Figure 8 illustrates distributed ART search in a system that is much like the one in Figure 7(a) except that F_2 has three coding nodes ($N = 3$). A distributed choice-by-difference signal rule sets the choice parameter $\alpha = 0.2$, a power law CAM rule sets $p = 1$, and input $\mathbf{a} = (0.7,0.8)$. For $j = 1,2$, thresholds are the same as the initial τ_{ij} values in Section 7.1. With Λ equal to the index set of above-average T_j, activity y_j is proportional to $T_j(1)$ when $T_j(1)$ is greater than or equal to the average $(\bar{T})$; otherwise $y_j = 0$. By hypothesis, $T_3(1) \leq 2.44$ so that initially, with all $\Delta_{ij} = \delta_{ij} = 0$, $T_1(1) = 3.20 > T_2(1) = 2.82 \geq \bar{T} > T_3(1)$. Thus, $y_1 = 0.532$ and $y_2 = 0.468$, as in Figure 7(a); and $y_3 = 0$. While this code y is active, the MTM depletion terms Δ_{ij} and δ_{ij} quickly go to equilibrium, sending the phasic terms $S_{ij}(y_j)$ [eqn (9)] and the tonic terms $\theta_{ij}(y_j)$ [eqn (13)] to 0. Then for $j = 1$, $(\Delta_{1j}...\Delta_{4j}) = (0.0,0.232,0.132,0.0)$, which will reduce $S_1(1)$ by 0.364 at reset; and $(\delta_{1j}...\delta_{4j}) = (0.532,0.3,0.4,0.532)$, which will reduce $\Theta_1(1)$ by 1.764 at reset. Thus, eqn (15) implies that, with $\alpha = 0.2$, $T_1(1)$ will be reduced by 1.78 at reset. For $j = 2$, $(\Delta_{1j}...\Delta_{4j}) = (0.0,0.0,0.0,0.2)$, which will reduce $S_2(1)$ by 0.2 at reset; and $(\delta_{1j}...\delta_{4j}) = (0.468,0.468,0.468,0.2)$, which will reduce $\theta_2(1)$ by 1.604 at reset. Thus, with $\alpha = 0.2$, $T_2(1)$ will be reduced by 1.48 at reset. Since $y_3 = 0$, Δ_{ij} and δ_{ij} remain equal to 0 for $j = 3$ and $i = 1...4$.

A reset with input **a** still active would then leave $T_3(1)$ unchanged but would reduce $T_1(1)$ from 3.20 to 1.42 and would reduce $T_2(1)$ from 2.82 to 1.34. What the next code y will be depends on the size of $T_3(1)$ (Table 2). When $T_3(1)$ is large ($1.5 < T_3(1) \leq 2.44$), node $j = 3$ is the only one active following reset, since $T_1(1)$ and $T_2(1)$ are then below average. With smaller $T_3(1)$ values ($1.38 \leq T_3(1) \leq 1.5$), nodes $j = 1$ and $j = 3$ share activation following reset. With even smaller $T_3(1)$ values ($1.26 < T_3(1) < 1.38$), $T_2(1)$ and $T_3(1)$ are below average, so node $j = 1$ is the only one active following reset. Finally, when $T_3(1)$ is very small ($0 \leq T_3(1) \leq 1.26$), nodes $j = 1$ and $j = 2$ share activation following reset, as they did before. However, the code y is now more uniform, with y_1 smaller and y_2 larger before the reset.

8. DISTRIBUTED ARTMAP

ARTMAP networks for supervised learning self-organize mappings from input vectors, representing features such as patient history and test results, to output vectors, representing predictions such as the likelihood of an adverse outcome following an operation. The original binary ARTMAP (Carpenter et al., 1991a) incorporates two ART 1 modules, ART$_a$ and ART$_b$, that are linked by a *map field* F^{ab}. At the map field the network forms associations between categories via outstar learning and triggers search, via the ARTMAP match tracking rule, when a training set input fails to make a correct prediction. Match tracking increases the ART$_a$ vigilance

TABLE 2

Distributed ART search in response to an input $a = (0.7,0.8)$, with complement coding, a power law CAM rule ($p = 1$) for above-average $T_j(1)$, a choice-by-difference signal rule ($\alpha = 0.2$), and $N = 3$

Before Reset: $T_1(1) = 3.20$, $T_2(1) = 2.82$					
	$T_3(1)$	y_1	y_2	y_3	$\bar{T}$
	$0 \leq T_3 \leq 2.44$	0.532	0.468	0	$2.01 \leq \bar{T} \leq 2.82$
After Reset: $T_1(1) = 1.42$, $T_2(1) = 1.34$					
	$T_3(1)$	y_1	y_2	y_3	$\bar{T}$
(a)	$1.5 < T_3 \leq 2.44$	0	0	1	$1.42 < \bar{T} \leq 1.73$
(b)	$1.38 \leq T_3 \leq 1.5$	$0.507 \geq 1.42/(1.42 + T_3) \geq 0.486$	0	$0.493 \leq T_3/(1.42 + T_3) \leq 0.514$	$1.38 \leq \bar{T} \leq 1.42$
(c)	$1.26 < T_3 < 1.38$	1	0	0	$1.34 < \bar{T} < 1.38$
(d)	$0 \leq T_3 \leq 1.26$	0.514	0.486	0	$0.92 \leq \bar{T} \leq 1.34$

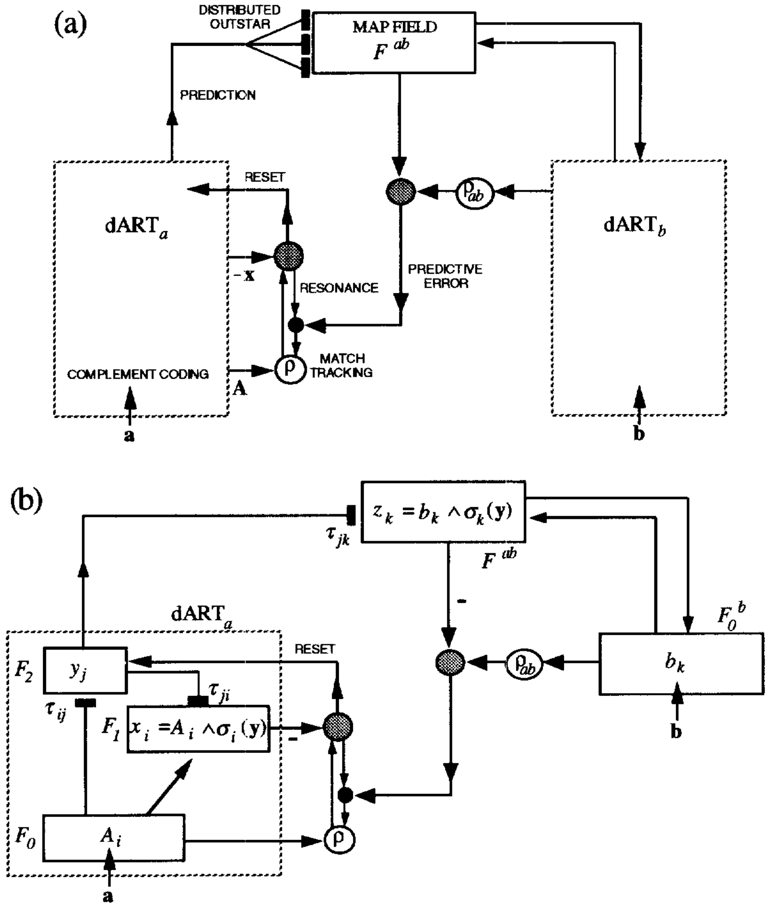

FIGURE 9. (a) Distributed ARTMAP substitutes dART modules for the ART modules in ARTMAP, and substitutes distributed outstar learning from ART$_a$ to the map field F^{ab} for outstar learning; (b) a simplified dARTMAP network computes classification probabilities, with $|\mathbf{b}| = 1$ at an output field F_0^b.

parameter ρ_a in response to a predictive error at ART$_b$. Fuzzy ARTMAP (Carpenter et al., 1992) substitutes fuzzy ART for ART 1. Distributed ARTMAP (dARTMAP) substitutes dART for fuzzy ART and distributed outstar learning for outstar learning at the map field [Figure 9(a)].

Many applications of supervised learning systems such as ARTMAP are classification problems, where the trained system tries to predict a correct category given a test set input vector. A prediction might be a single category or distributed as a set of scores or probabilities. For this class of problems, the dARTMAP architecture illustrated in Figure 9(b) does not require the full dART$_b$ architecture. Even in this case, however, dARTMAP implementation requires a number of judicious design choices, in contrast to the few choices required for fuzzy ARTMAP implementation. Recent benchmark simulation studies have demonstrated that, with fast learning and noisy training data, dARTMAP maintains the predictive accuracy of fuzzy ARTMAP while dramatically improving code compression. Ongoing research seeks to characterize how a distributed learning system such as dARTMAP can combine speed, performance, generalization, and code compression in a variety of new applications.

REFERENCES

Bachelder, I. A., Waxman, A. M., & Seibert, M. (1993). A neural system for mobile robot visual place learning and recognition. In *Proceedings of the World Congress on Neural Networks (WCNN'93)* (pp. 512–517). Hillsdale, NJ: Erlbaum.

Baloch, A. A., & Waxman, A. M. (1991). Visual learning, adaptive expectations, and behavioral conditioning of the mobile robot MAVIN. *Neural Networks, 4,* 271–302.

Baraldi, A., & Parmiggiani, F. (1995). A neural network for unsupervised categorization of multivalued input patterns, an application of satellite image clustering. *IEEE Transactions on Geoscience and Remote Sensing, 33,* 305–316.

Bernardon, A. M., & Carrick, J. E. (1995). A neural system for automatic target learning and recognition applied to bare and camouflaged SAR targets. *Neural Networks, 8,* 1103–1108.

Carpenter, G. A. (1994a). A distributed outstar network for spatial pattern learning. *Neural Networks,* **7**, 159–168.

Carpenter, G. A. (1994b). Distributed recognition codes and catastrophic forgetting. In *Proceedings of the World Congress on Neural Networks (WCNN'94)* (pp. 133–142). Hillsdale, NJ: Erlbaum.

Carpenter, G. A., & Gjaja, M. N. (1994). Fuzzy ART choice functions.

In *Proceedings of the World Congress on Neural Networks (WCNN'94)* (pp. 713–722). Hillsdale, NJ: Erlbaum.

Carpenter, G. A., & Grossberg, S. (1987). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing, 37,* 54–115.

Carpenter, G. A., & Grossberg, S. (1991). *Pattern recognition by self-organizing neural networks.* Cambridge, MA: MIT Press.

Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J. H., & Rosen, D. B. (1992). Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks, 3,* 698–713.

Carpenter, G. A., Grossberg, S., & Reynolds, J. H. (1991a). ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. *Neural Networks,* **4**, 565–588.

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

Carpenter, G. A., & Markuzon, N. (1996). *ARTMAP-IC and medical diagnosis: Instance counting and inconsistent cases* (CAS/CNS Tech. Report CAS/CNS-96-017). Boston, MA: Boston University.

Carpenter, G. A., & Ross, W. D. (1993). ART-EMAP: A neural network architecture for learning and prediction by evidence accumulation. In *Proceedings of the World Congress on Neural Networks (WCNN'94)* (pp. 649–656). Hillsdale, NJ: Erlbaum.

Carpenter, G. A., & Ross, W. D. (1995). ART-EMAP: A neural network architecture for object recognition by evidence accumulation. *IEEE Transactions on Neural Networks, 6,* 805–818.

Caudell, T. P., & Healy, M. J. (1994). Adaptive Resonance Theory networks in the Encephalon autonomous vision system. In *Proceedings of the 1994 IEEE International Conference on Neural Networks (ICNN'94)* (pp. 1235–1240). Piscataway, NJ: IEEE.

Caudell, T. P., Smith, S. D. G., Escobedo, R., & Anderson, M. (1994). NIRS: Large scale ART-1 neural architectures for engineering design retrieval. *Neural Networks, 7,* 1339–1350.

Christodoulou, C. G., Huang, J., Georgiopoulos, M., & Liou, J. J. (1995). Design of gratings and frequency selective surfaces using fuzzy ARTMAP neural networks. *Journal of Electromagnetic Waves and Applications, 9,* 17–36.

Dubrawski, A., & Crowley, J. L. (1994). Learning locomotion reflexes: A self-supervised neural system for a mobile robot. *Robotics and Autonomous Systems, 12,* 133–142.

Gan, K. W., & Lua, K. T. (1992). Chinese character classification using an adaptive resonance network. *Pattern Recognition, 25,* 877–888.

Gjerdingen, R. O. (1990). Categorization of musical patterns by self-organizing neuronlike networks. *Music Perception, 7,* 339–370.

Gopal, S., Sklarew, D. M., & Lambin, E. (1994). Fuzzy-neural networks in multi-temporal classification of landcover change in the Sahel. In *Proceedings of the DOSES Workshop on New Tools for Spatial Analysis* (pp. 55–68). Lisbon, Portugal: DOSES, EUROSTAT; Brussels, Luxembourg: ECSC-EC-EAEC.

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

Grossberg, S. (1970). Some networks that can learn, remember, and reproduce any number of complicated space–time patterns. *Studies in Applied Mathematics, 49,* 135–166.

Grossberg, S. (1972). Neural expectation: Cerebellar and retinal analogs of cells fired by learnable or unlearned pattern classes. *Kybernetik, 10,* 49–57.

Grossberg, S. (1976a). Adaptive pattern classification and universal recoding. II: Feedback, expectation, olfaction, and illusions. *Biological Cybernetics,* **23**, 187–202.

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

Grossberg, S. (1980). How does a brain build a cognitive code?. *Psychological Review, 87,* 151.

Ham, F. M., & Han, S. (1996). Classification of cardiac arrhythmias using fuzzy ARTMAP. *IEEE Transactions on Biomedical Engineering, 43,* 425–430.

Kalkunte, S. S., Kumar, J. M., & Patnaik, L. M. (1992). A neural network approach for high resolution fault diagnosis in digital circuits. In *Proceedings of the International Joint Conference on Neural Networks* (pp. 83–88). Piscataway, NJ: IEEE.

Kasperkiewicz, J., Racz, J., & Dubrawski, A. (1995). HPC strength prediction using artificial neural network. *Journal of Computing in Civil Engineering, 9,* 279–284.

Kim, J. W., Jung, K. C., Kim, S. K., & Kim, H. J. (1995). Shape classification of on-line Chinese character strokes using ART 1 neural network. In *Proceedings of the World Congress on Neural Networks (WCNN'95)* (pp. 191–194). Hillsdale, NJ: Erlbaum.

Koch, M. W., Moya, M. M., Hostetler, L. D., & Fogler, R. J. (1995). Cueing, feature discovery, and one-class learning for synthetic aperture radar automatic target recognition. *Neural Networks, 8,* 1081–1102.

Ly, S., & Choi, J. J. (1994). Drill condition monitoring using ART-1. In *Proceedings of the IEEE International Conference on Neural Networks (ICNN'94)* (pp. 1226–1229). Piscataway, NJ: IEEE.

Malsburg, C. (1973). Self-organization of orientation sensitive cells in the striate cortex. *Kybernetik, 14,* 85–100.

Markram, H., & Tsodyks, M. (1996). Redistribution of synaptic efficacy between neocortical pyramidal neurons. *Nature, 382,* 807–810.

Mehta, B. V., Vij, L., & Rabelo, L. C. (1993). Prediction of secondary structures of proteins using fuzzy ARTMAP. In *Proceedings of the World Congress on Neural Networks (WCNN'93)* (pp. 228–232). Hillsdale, NJ: Erlbaum.

Murshed, N. A., Bortozzi, F., & Sabourin, R. (1995). Off-line signature verification, without *a priori* knowledge of class ω2. A new approach. In *Proceedings of the Third International Conference on Document Analysis and Recognition (ICDAR'95),* Piscataway, NJ: IEEE.

Nicholls, D. G. (1994). *Proteins, transmitters and synapses.* Oxford: Blackwell.

Racz, J., & Dubrawski, A. (1995). Artificial neural network for mobile robot topological localization. *Robotics and Autonomous Systems, 16,* 73–80.

Rubin, M. A. (1995). Application of fuzzy ARTMAP and ART-EMAP to automatic target recognition using radar range profiles. *Neural Networks, 8,* 1109–1116.

Seibert, M., & Waxman, A. M. (1992). Adaptive 3D object recognition from multiple views. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 14,* 107–124.

Seibert, M., & Waxman, A. M. (1993). An approach to face recognition using saliency maps and caricatures. In *Proceedings of the World Congress on Neural Networks (WCNN'93)* (pp. 661–664). Hillsdale, NJ: Erlbaum.

Soliz, P., & Donohoe, G. W. (1996). Adaptive resonance theory neural network for fundus image segmentation. In *Proceedings of the World Congress on Neural Networks (WCNN'96)* (pp. 1180–1183). Hillsdale, NJ: Erlbaum.

Srinivasa, N., & Sharma, R. (1996). A self-organizing invertible map for active vision applications. In *Proceedings of the World Congress on Neural Networks (WCNN'96)* (pp. 121–124). Hillsdale, NJ: Erlbaum.

Suzuki, Y. (1995). Self-organizing *QRS*-wave recognition in ECG using neural networks. *IEEE Transactions on Neural Networks, 6,* 1469–1477.

Tarng, Y. S., Li, T. C., & Chen, M. C. (1994). Tool failure monitoring for drilling processes. In *Proceedings of the 3rd International Conference on Fuzzy Logic, Neural Nets and Soft Computing* (pp. 109–111). Iizuka, Japan.

Tse, P., & Wang, D. D. (1996). A hybrid neural networks based machine condition forecaster and classifier by using multiple vibration parameters. In *Proceedings of the 1994 IEEE International Conference on Neural Networks* (pp. 2096–2100). Piscataway, NJ: IEEE.

Waxman, A. M., Seibert, M. C., Gove, A., Fay, D. A., Bernardon, A. M., Lazott, C., Steele, W. R., & Cunningham, R. K. (1995). Neural processing of targets in visible, multispectral IR and SAR imagery. *Neural Networks, 8,* 1029–1051.

Wienke, D. (1994). Neural resonance and adaption—towards nature's principles in artificial pattern recognition. In L. Buydens and W. Melssen (Eds.), *Chemometrics: Exploring and exploiting chemical information.* Nijmegen, NL: University Press.

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