ART and ARTMAP Neural Networks for Applications: Self-Organizing Learning, Recognition, and Prediction

Gail A. Carpenter*, Marin N. Gjaja*, Sucharita Gopal**, Natalya Markuzon*, Curtis E. Woodcock**

*Center for Adaptive Systems
and
Department of Cognitive and Neural Systems

**Center for Remote Sensing
and
Department of Geography

Boston University
Boston, Massachusetts 02215 USA

To appear in: *Intelligent Systems in Engineering: Approaches and Applications*
Lakhmi C. Jain, Editor
Cambridge, Massachusetts: MIT Press
March, 1996

Technical Report CAS/CNS TR-96-009
Boston, MA: Boston University

This research was supported in part by the National Science Foundation (NSF IRI 94-01659 and NSF SBR 93-00633) and the Office of Naval Research (ONR N00014-95-1-0409 and ONR N00014-95-0657).

ART and ARTMAP Neural Networks for Applications: Self-Organizing Learning, Recognition, and Prediction

TABLE OF CONTENTS

SUMMARY

ART and ARTMAP Neural Networks for Applications: Self-Organizing Learning, Recognition, and Prediction

ART and ARTMAP neural networks for adaptive recognition and prediction have been applied to a variety of problems. Applications include parts design retrieval at the Boeing Company, automatic mapping from remote sensing satellite measurements, medical database prediction, and robot vision. This chapter features a self-contained introduction to ART and ARTMAP dynamics and a complete algorithm for applications. Computational properties of these networks are illustrated by means of remote sensing and medical database examples. The basic ART and ARTMAP networks feature winner-take-all (WTA) competitive coding, which groups inputs into discrete recognition categories. WTA coding in these networks enables fast learning, that allows the network to encode important rare cases but that may lead to inefficient category proliferation with noisy training inputs. This problem is partially solved by ART-EMAP, which use WTA coding for learning but distributed category representations for test-set prediction. In medical database prediction problems, which often feature inconsistent training input predictions, the ARTMAP-IC network further improves ARTMAP performance with distributed prediction, category instance counting, and a new search algorithm. A recently developed family of ART models (dART and dARTMAP) retains stable coding, recognition, and prediction, but allows arbitrarily distributed category representation during learning as well as performance.

1. ART AND ARTMAP NEURAL NETWORKS

Adaptive resonance theory originated from an analysis of human cognitive information processing and stable coding in a complex input environment (Grossberg, 1976, 1980). An evolving series of ART neural network models have added new principles to the early theory and have realized these principles as quantitative systems that can be applied to problems of category learning, recognition, and prediction. Each ART network forms stable recognition categories in response to arbitrary input sequences with either fast or slow learning regimes (Section 2). The first ART model, ART 1 (Carpenter and Grossberg, 1987a), was an unsupervised learning system to categorize binary input patterns. ART 2 (Carpenter and Grossberg, 1987b) and fuzzy ART (Carpenter, Grossberg, and Rosen, 1991) extend the ART 1 domain to categorize analog as well as binary input patterns (Sections 3-5) (Carpenter & Grossberg, 1991).

Supervised ART architectures, called ARTMAP systems, self-organize arbitrary mappings from input vectors, representing features such as spectral values and terrain variables, to output vectors, representing predictions such as vegetation classes in a remote sensing application (Section 6). Internal ARTMAP control mechanisms create stable recognition categories of optimal size by maximizing code compression while minimizing predictive error in an on-line setting. Binary ART 1 computations are the foundation of the first ARTMAP network (Carpenter, Grossberg, and Reynolds, 1991), which therefore learns binary maps. When fuzzy ART replaces ART 1 in an ARTMAP system, the resulting fuzzy ARTMAP architecture (Carpenter, Grossberg, Markuzon, Reynolds, & Rosen, 1992) rapidly learns stable mappings between analog or binary input and output vectors. Section 7 includes a complete fuzzy ARTMAP implementation algorithm for applications.

Recently fuzzy ARTMAP has become the basis of new methodologies for producing maps from satellite data (Carpenter, Gjaja, Gopal, & Woodcock, 1995; Gopal, Sklarew, & Lambin, 1994). A simplified version of this problem (Section 8) introduces and illustrates the dynamics of fuzzy ARTMAP networks. A medical database prediction example (Section 9) illustrates how the basic ARTMAP system can be augmented to meet the computational demands of particular classes of supervised learning problems. Other applications of unsupervised ART networks and supervised ARTMAP networks include a Boeing parts design retrieval system (Caudell, Smith, Escobedo, & Anderson, 1994), robot sensory-motor control (Bachelder, Waxman, & Seibert, 1993; Baloch & Waxman, 1991; Dubrawski & Crowley, 1994a), robot navigation (Dubrawski & Crowley, 1994b), machine vision (Caudell & Healy, 1994), 3D object recognition (Seibert & Waxman, 1992), face recognition (Seibert & Waxman, 1993), Macintosh operating system software (Johnson, 1993), automatic target recognition (Bernardon & Carrick, 1995; Koch, Moya, Hostetler, & Fogler, 1995; Waxman et al., 1995), electrocardiogram wave recognition (Ham & Han, 1993; Suzuki, Abe, & Ono, 1993), prediction of protein secondary structure (Mehta, Vij, & Rabelo, 1993), air quality monitoring (Wienke, Xie, & Hopke, 1994), strength prediction for concrete mixes (Kasperkiewicz, Racz, & Dubrawski, 1994), signature verification (Murshed, Bortozzi, & Sabourin, 1995), tool failure monitoring (Ly & Choi, 1994; Tarng, Li, & Chen, 1994), chemical analysis from UV and IR spectra (Wienke & Kateman, 1994), frequency selective surface design for electromagnetic system devices (Christodoulou, Huang, Georgiopoulos, & Liou, 1995), Chinese character recognition (Gan & Lua, 1992), and analysis of musical scores (Gjerdingen, 1990).

2. ART DYNAMICS

The central feature of all ART systems is a pattern matching process that compares the current input with a learned category representation, or active hypothesis, selected by the input.. This matching process leads either to a resonant state that focuses attention and triggers category learning or to a self-regulating parallel memory search that always leads to a resonant sate, unless the network's memory capacity is exceeded. If the search ends with selection of an established category, then the category's learned representation may be refined to incorporate new information from the current input. If the search ends by selecting a previously untrained node, the ART network establishes a new category.

Figure 1 illustrates the main components of an ART 1 network and Figure 2 illustrates the ART search cycle. During ART search, an input vector **A** registers itself as a pattern **x** of activity across level F_1 (Figure 2a). Converging and diverging $F_1 \rightarrow F_2$ adaptive filter pathways, each weighted by a long term memory (LTM) trace, or adaptive weight, transform **x** into a net input vector **T** to level F_2. The internal competitive dynamics of F_2 contrast-enhance vector **T**, generating a compressed activity vector **y** across F_2. In ART 1 and fuzzy ART, strong competition selects the F_2 node that receives the maximal $F_1 \rightarrow F_2$ input component T_J Only one component $\left(y_J\right)$ of **y** remains positive after this choice takes place. Activation of such a winner-take-all node selects category J for the input pattern **A**.

Activation of an F_2 node may be interpreted as "making a hypothesis" about an input **A**. After sending the F_2 activity vector **y** through top-down adaptive filter pathways, a filtered vector **V** becomes the $F_2 \rightarrow F_1$ input (Figure 2b). The ART network matches the "expectation" pattern **V** of the active category against the current input pattern, or exemplar, **A**. This matching process typically changes the F_1 activity pattern **x**, suppressing activation of all features in **A** that are not confirmed by **V**. The resultant pattern $\mathbf{x}^*$ represents the features to which the network "pays attention." If the expectation **V** is close enough to the input **A**, then a state of resonance occurs, with the matched pattern $\mathbf{x}^*$ defining an attentional focus. The resonant state persists long enough for weight adaptation to occur; hence the term *adaptive resonance* theory. The fact that ART networks encode only attended features $\mathbf{x}^*$ rather than all input features **A** is directly responsible for ART code stability.

A dimensionless parameter called *vigilance* defines the criterion of an acceptable match. Vigilance specifies what fraction of the bottom-up input **A** must remain in the matched F_1 pattern $\mathbf{x}^*$ in order for resonance to occur. In ARTMAP, vigilance becomes an internally controlled variable, rather than the fixed parameter of ART. Because vigilance then varies across learning trials, a single ARTMAP system can encode widely differing degrees of generalization, or code compression. Low vigilance allows broad generalization, coarse categories, and abstract representations. High vigilance leads to narrow generalization, fine categories, and specific representations. At the very high vigilance limit, category learning reduces to exemplar learning. Varying vigilance levels allow a single ART system to recognize both abstract categories, such as faces and dogs, and individual faces and dogs.

ART memory search, or hypothesis testing, begins when the top-down expectation **V** determines that the bottom-up input **A** is too novel, or unexpected, with respect to the chosen category to satisfy the vigilance criterion. Search leads to selection of a better recognition code to represent input **A** at level F_2. An *orienting subsystem* Ω controls the search process. The orienting subsystem interacts with the attentional subsystem, as in Figures 2b and 2c, to enable the network to learn about novel inputs without risking unselective forgetting of its previous knowledge. ART 3 (Carpenter & Grossberg, 1990) implements parallel distributed search as a medium-term memory (MTM) process, as needed for distributed recognition codes.

ART search prevents associations from forming between **y** and **x*** if **x*** is too different from **A** to satisfy the vigilance criterion. The search process resets **y** before such an association can form. If the search ends upon a familiar category, then that category's representation may be refined in light of new information carried by **A**. If the search ends upon an uncommitted F_2 node, then **A** begins a new category. An ART *choice parameter* controls how deeply the search proceeds before selecting an uncommitted node. In a parameter range called the *conservative limit*, the choice parameter α is very small. Then an input first selects a category whose weight vector is a subset of the input, if such a category exists. Given such a choice, no weight change occurs during learning; hence the name conservative limit, since learned weights are conserved wherever possible. As learning self-stabilizes, all inputs coded by a category access it directly and search is automatically disengaged.

3. FUZZY ART

The ART 1 operations of category choice, matching, and learning translate into fuzzy ART operations when the intersection operator ($\cap$) of ART 1 is replaced by the fuzzy intersection, or component-wise minimum, operator ($\wedge$). For the special case of binary inputs and fast learning, the computations of fuzzy ART are identical to those of ART 1.

Many ART applications use fast learning, whereby adaptive weights fully converge to equilibrium values in response to each input pattern. Fast learning enables a system to adapt quickly to inputs that occur only rarely but that may require immediate accurate performance. Remembering many details of an exciting movie is a typical example of fast learning. Fast learning destabilizes the memories of feedforward, error-based models like back propagation. When the difference between actual output and target output defines "error," present inputs drive out past learning, since fast learning zeroes the error on each input trial. This feature of back propagation typically restricts its domain to off-line applications with a slow learning rate. In addition, lacking the key feature of competition, a back propagation system tends to average rare events with similar frequent events that have different consequences.

Some applications benefit from a *fast-commit slow-recode* option that combines fast initial learning with a slower rate of forgetting. Fast commitment retains the advantage of fast learning, namely, the ability to respond to important distinctive inputs that occur only rarely. Slow recoding then prevents features in a category's learned representation from being erroneously altered in response to noisy or partial inputs.

Complement coding is a preprocessing step that normalizes input patterns and solves a potential fuzzy ART category proliferation problem (Carpenter, Grossberg, & Rosen, 1991). In neurobiological terms, complement coding uses both on-cells and off-cells to represent an input pattern, preserving individual feature amplitudes while normalizing the total on-cell/off-cell activity. Functionally, the on-cell portion of a weight vector encodes features that are consistently present in category exemplars, while the off-cell portion encodes features that are consistently absent. Small weights in both on-cell and off-cell portions of a category representation encode as "uninformative" those features that are sometimes present and sometimes absent. Complement coding allows a geometric interpretation of fuzzy ART recognition categories as box-shaped regions of input space. Simulations of a prototype remote sensing example illustrate fuzzy ART geometry with inputs that provide two TM spectral band values at each pixel (Section 8). Thus the inputs are two-dimensional and category boxes are rectangles.

4. FUZZY ART DYNAMICS

This section summarizes the dynamics of a fuzzy ART network with a complement coding preprocessor. Each step is found in the summary fuzzy ARTMAP algorithm (Section 7), which includes a fuzzy ART algorithm as a special case, just as a fuzzy ART network is embedded in the simplified ARTMAP network shown in Figure 3.

4.1 - Field Activity Vectors

A fuzzy ART system includes a field F_0 of nodes that represent a current input vector; a field F_2 that represents the active code, or category; and a field F_1 that receives both bottom-up input from F_0 and top-down input from F_2 (Figure 3). Vector $\mathbf{A}$ denotes F_0 activity, with each component A_i in the interval $[0,1]$. With complement coding, $\mathbf{A}=\left(\mathbf{a},\mathbf{a}^c\right)$. That is, $A_i=a_i$ for $i=1...M$; and $A_i=a_{i-M}^c\equiv\left(1-a_{i-M}\right)$ for $i=M+1...2M$. Vector $\mathbf{x}=\left(x_1,...,x_{2M}\right)$ denotes F_1 activity and $\mathbf{y}=\left(y_1,...,y_N\right)$ denotes F_2 activity. The number of input components (M) and the number of category nodes (N) can be arbitrarily large..

4.2 - Weight Vector

Associated with each F_2 category node j $(j=1...N)$ is a vector $\mathbf{w}_j\equiv\left(w_{j1},...,w_{j,2M}\right)$ of adaptive weights, or long-term memory (LTM) traces. Initially:

$$w_{j1}(0)=...=w_{j,2M}(0)=1. \qquad \text{(eq 1)}$$

Then each category is *uncommitted*. After a category codes its first input, it becomes *committed*. Each component w_{ji} can decrease toward 0 but never increase during learning, so weights always converge during learning. The fuzzy ART weight vector $\mathbf{w}_j$ denotes both the bottom-up and top-down weight vectors.

4.3 - Parameters

A choice parameter $\alpha>0$, a learning rate parameter $\beta\in[0,1]$, and a vigilance parameter $\rho\in[0,1]$ determine fuzzy ART dynamics.

4.4 - Category Choice

For each input **a** and F_2 node *j*, a Weber law *choice function* T_j is defined by

$$T_j=\frac{\left|\mathbf{A}\wedge\mathbf{w}_j\right|}{\alpha+\left|\mathbf{w}_j\right|}, \qquad \text{(eq 2)}$$

where the fuzzy intersection $\wedge$ (Zadeh, 1965) is defined by

$$(\mathbf{p}\wedge\mathbf{q})_i\equiv\min\left(p_i,q_i\right) \qquad \text{(eq 3)}$$

and where the city-block norm $|\cdots|$ is defined by

$$|\mathbf{p}| \equiv \sum_i |p_i|. \qquad \text{(eq 4)}$$

Alternative choice functions (Section 7.1, Step (2)) include choice-by-difference (Carpenter & Gjaja, 1994), which selects the categories to minimize weight changes.

The system makes a *category choice* when at most one F_2 node can become active at a given time. The index J denotes the chosen category, where

$$T_J = \max\left\{T_j : j = 1, \ldots, N\right\}. \qquad \text{(eq 5)}$$

If more than one T_j is maximal, the category with the smallest j index is chosen. In particular, nodes become committed in order $j = 1,2,3\ldots$. When the J^{th}category is chosen, $y_J = 1$; and $y_j = 0$ for $j \neq J$. The $F_2 \to F_1$ signal vector $\mathbf{V}$ is then equal to the J^{th} category weight vector $\mathbf{w}_J$ and the F_1 activity vector $\mathbf{x}$ is reduced from $\mathbf{A}$ to the matched pattern $\mathbf{A} \wedge \mathbf{w}_J$. That is, in a choice system, the F_1 vector $\mathbf{x}$ obeys the equation

$$\mathbf{x} = \begin{cases} \mathbf{A} & \text{if } F_2 \text{ is inactive} \\ \mathbf{A} \wedge \mathbf{w}_J & \text{if the } J^{th}\ F_2 \text{ node is chosen.} \end{cases} \qquad \text{(eq 6)}$$

4.5 - Resonance or Reset

Resonance occurs if the *match function* $\left|\mathbf{A} \wedge \mathbf{w}_J\right| |\mathbf{A}|^{-1}$ of the chosen category meets the vigilance criterion:

$$\frac{\left|\mathbf{A} \wedge \mathbf{w}_J\right|}{|\mathbf{A}|} \geq \rho; \qquad \text{(eq 7)}$$

that is, by (eq 6), when the J^{th} category becomes active, resonance occurs if

$$|\mathbf{x}| = \left|\mathbf{A} \wedge \mathbf{w}_J\right| \geq \rho|\mathbf{A}|. \qquad \text{(eq 8)}$$

Learning then ensues, as defined below. *Mismatch reset* occurs if

$$\frac{\left|\mathbf{A} \wedge \mathbf{w}_J\right|}{|\mathbf{A}|} < \rho; \qquad \text{(eq 9)}$$

that is, if

$$|\mathbf{x}| = \left|\mathbf{A} \wedge \mathbf{w}_J\right| < \rho|\mathbf{A}|. \qquad \text{(eq 10)}$$

Then the value of the choice function T_J is set to 0 for the duration of the input presentation to prevent the persistent selection of the same category during search. A new index J represents the active category, selected by (eq 5). The search process continues until the chosen J satisfies the matching criterion (eq 7). By (eq 1), search ends if J is an uncommitted node.

4.6 - Learning

Once search ends, the weight vector $\mathbf{w}_J$ learns according to the equation

$$\mathbf{w}_J^{(\text{new})} = \beta\left(\mathbf{A} \wedge \mathbf{w}_J^{(\text{old})}\right) + (1-\beta)\mathbf{w}_J^{(\text{old})} \qquad \text{(eq 11)}$$

(Moore, 1989). *Fast learning* corresponds to setting $\beta = 1$. The weight vector $\mathbf{w}_J$ then converges to the matched F_1 vector $\mathbf{x} = \mathbf{A} \wedge \mathbf{w}_J$ on each input presentation.

4.7 - Normalization by Complement Coding

Normalization of fuzzy ART inputs prevents category proliferation as many weights erode to 0 in some cases. An $F_0 \rightarrow F_1$ input is normalized if $\sum_{i=1}^{2M} A_i = |\mathbf{A}| \equiv$ constant for all inputs $\mathbf{A}$. Complement coded inputs are automatically normalized because

$$|\mathbf{A}| = \left|\left(\mathbf{a}, \mathbf{a}^c\right)\right| = \sum_{i=1}^{M} a_i + \sum_{i=1}^{M} (1 - a_i) = M. \qquad \text{(eq 12)}$$

5. FUZZY ART GEOMETRY

A geometric interpretation of fuzzy ART represents each category as a box in M-dimensional space, where M is the number of components of input $\mathbf{a}$. In the prototype remote sensing example (Section 8), $\mathbf{a}$ represents two Landsat Thematic Mapper (TM) spectral band values for a given pixel, scaled to the interval $[0,1]$, so $M = 2$. With complement coding, then,

$$\mathbf{A} = \left(\mathbf{a}, \mathbf{a}^c\right) \equiv \left(a_1, a_2, a_1^c, a_2^c\right). \qquad \text{(eq 13)}$$

With $M = 2$, each category j has a geometric representation as a rectangle R_j. Following the form of equation (eq 13), a complement-coded weight vector $\mathbf{w}_j$ can be written as

$$\mathbf{w}_j = \left(\mathbf{u}_j, \mathbf{v}_j^c\right), \qquad \text{(eq 14)}$$

where $\mathbf{u}_j$ and $\mathbf{v}_j$ are 2-dimensional vectors. Vector $\mathbf{u}_j$ defines one corner of a rectangle R_j and $\mathbf{v}_j$ defines the opposite corner (Figure 4a). The size of R_j is

$$\left|R_j\right| \equiv \left|\mathbf{v}_j - \mathbf{u}_j\right|, \qquad \text{(eq 15)}$$

which is equal to the height plus the width of R_j. In the prototype example, each side of R_j represents a range of values of the corresponding TM band.

In a fast-learn fuzzy ART system, with $\beta = 1$ (eq 11), $\mathbf{w}_J^{(\text{new})} = \mathbf{A} = \left(\mathbf{a}, \mathbf{a}^c\right)$ when J is an uncommitted node. The corners of $R_J^{(\text{new})}$ are then $\mathbf{a}$ and $\left(\mathbf{a}^c\right)^c = \mathbf{a}$. Hence $R_J^{(\text{new})}$ is just the point box $\mathbf{a}$. Learning increases the size of R_J, which grows as weights shrink. Vigilance ρ determines the maximum size of R_j, with $|R_j| \le M(1-\rho)$, as shown below. With fast learning, R_J expands to $R_J \oplus \mathbf{a}$, the minimum box containing R_J and $\mathbf{a}$ (Figure 4b). The corners of $R_J \oplus \mathbf{a}$ are $\mathbf{a} \wedge \mathbf{u}_J$ and $\mathbf{a} \vee \mathbf{v}_J$, where the fuzzy intersection $\wedge$ is defined by (eq 3); and the fuzzy union $\vee$ is defined by

$$(\mathbf{p} \vee \mathbf{q})_i \equiv \max(p_i, q_i) \qquad \text{(eq 16)}$$

(Zadeh, 1965). Hence, by (eq 15), the size of $R_J \oplus \mathbf{a}$ is:

$$|R_J \oplus \mathbf{a}| = |(\mathbf{a} \vee \mathbf{v}_J) - (\mathbf{a} \wedge \mathbf{u}_J)|. \qquad \text{(eq 17)}$$

However, before R_J can expand to include $\mathbf{a}$, category J is reset if $|R_J \oplus \mathbf{a}|$ would be too large, according to the vigilance criterion. With fast learning, R_j is the smallest box that encloses all vectors $\mathbf{a}$ that have chosen category j without reset.

If $\mathbf{a}$ has dimension M, the box R_j includes the two opposing vertices $\wedge_j \mathbf{a}$ and $\vee_j \mathbf{a}$, where the i^{th} component of each of these vectors is:

$$\left(\wedge_j \mathbf{a}\right)_i = \min\{a_i : \mathbf{a} \text{ has been coded by category } j\} \qquad \text{(eq 18)}$$

and

$$\left(\vee_j \mathbf{a}\right)_i = \max\{a_i : \mathbf{a} \text{ has been coded by category } j\} \qquad \text{(eq 19)}$$

(Figure 4c). The size of R_j is

$$|R_j| = |\vee_j \mathbf{a} - \wedge_j \mathbf{a}| \qquad \text{(eq 20)}$$

and the weight vector $\mathbf{w}_j$ is

$$\mathbf{w}_j = \left(\wedge_j \mathbf{a}, \left(\vee_j \mathbf{a}\right)^c\right), \qquad \text{(eq 21)}$$

as in (eq 14) and (eq 15). Thus

$$|\mathbf{w}_j| = \sum_i \left(\wedge_j \mathbf{a}\right)_i + \sum_i \left[1 - \left(\vee_j \mathbf{a}\right)_i\right] = M - |\vee_j \mathbf{a} - \wedge_j \mathbf{a}|, \qquad \text{(eq 22)}$$

so the size of the box R_j is

$$|R_j| = M - |\mathbf{w}_j|. \qquad \text{(eq 23)}$$

By (eq 8), (eq 11), and (eq 12),

$$\left|\mathbf{w}_j\right| \geq \rho M. \qquad \text{(eq 24)}$$

By (eq 23) and (eq 24),

$$\left|R_j\right| \leq (1-\rho)M. \qquad \text{(eq 25)}$$

Inequality (eq 25) shows that high vigilance ($\rho \cong 1$) leads to small R_j while low vigilance ($\rho \cong 0$) permits large R_j.

6. 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, Grossberg, & Reynolds, 1991) incorporates two ART 1 modules, ART$_a$ and *ART$_b$*, that are linked by a *map field* F^{ab} (Figure 5). During supervised learning, ART$_a$ receives a stream of patterns $\left\{\mathbf{a}^{(n)}\right\}$ and ART$_b$ receives a stream of patterns $\left\{\mathbf{b}^{(n)}\right\}$, where $\mathbf{b}^{(n)}$ is the correct prediction given $\mathbf{a}^{(n)}$. An associative learning network and an internal controller link these modules to make the ARTMAP system operate in real time. The controller creates the minimal number of ART$_a$ recognition categories, or "hidden units," needed to meet accuracy criteria. A minimax learning rule enables ARTMAP to learn quickly, efficiently, and accurately as it conjointly minimizes predictive error and maximizes code compression. This scheme automatically links predictive success to category size on a trial-by-trial basis using only local operations. It works by increasing the ART$_a$ vigilance parameter ρ_a by the minimal amount needed to correct a predictive error at ART$_b$.

At the map field an ARTMAP network forms associations between categories via outstar learning and triggers search, via a *match tracking* rule, when a training set input fails to make a correct prediction. Match tracking increases the ART$_a$ vigilance parameter ρ_a in response to a predictive error at *ART$_b$*. A *baseline vigilance* parameter $\overline{\rho}_a$ calibrates a minimum confidence level at which ART$_a$ will accept a chosen category. Lower values of $\overline{\rho}_a$ allow larger categories to form, maximizing code compression. Initially, $\rho_a = \overline{\rho}_a$. During training, a predictive failure at ART$_b$ increases ρ_a just enough to trigger an ART$_a$ search. Match tracking sacrifices the minimum amount of compression necessary to correct the predictive error. Hypothesis testing selects a new ART category, which focuses attention on a cluster of $\mathbf{a}^{(n)}$ input features that is better able to predict $\mathbf{b}^{(n)}$. With fast learning, match tracking allows a single ARTMAP system to learn a different prediction for a rare event than for a cloud of similar frequent events in which it is embedded. Fuzzy ARTMAP (Carpenter, Grossberg, Markuzon, Reynolds, & Rosen, 1992) substitutes fuzzy ART for ART 1.

7. A FUZZY ARTMAP ALGORITHM

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. The fuzzy ARTMAP algorithm below outlines a procedure for these problems, which do not require the full ART$_b$ architecture. The algorithm implements a fuzzy ARTMAP network (Figure 3) that is a simplified version of the full network (Figure 5) but that nevertheless is sufficient for most current applications. In the algorithm an input $\mathbf{a} = (a_1 \ldots a_i \ldots a_M)$ learns to predict an outcome $\mathbf{b} = (b_1 \ldots b_k \ldots b_L)$. A classification problem would set one component $b_K = 1$ during training, placing an input $\mathbf{a}$ in class K.

Note that the algorithm allows a small match-tracking parameter (ε) to be either positive or negative. Compared to the original match tracking algorithm, which allowed only positive ε values, a negative value of ε can facilitate prediction with sparse or inconsistent data and improve memory compression without loss of accuracy, and the resulting algorithm is actually a better approximation of the full ARTMAP differential equations (Carpenter & Markuzon, 1996).

7.1 - Fuzzy ARTMAP Training

During training, input pairs $\left(\mathbf{a}^{(1)}, \mathbf{b}^{(1)}\right), \left(\mathbf{a}^{(2)}, \mathbf{b}^{(2)}\right), \ldots, \left(\mathbf{a}^{(n)}, \mathbf{b}^{(n)}\right), \ldots$ are presented for equal time intervals. Each ART$_a$ input is complement coded, with $0 \le a_i \le 1$, $a_i^c \equiv 1 - a_i$, and input $\mathbf{A} = \left(\mathbf{a}, \mathbf{a}^c\right)$, so $|\mathbf{A}| \equiv M$ The output $\mathbf{b}$ is normalized to 1 $\left(\sum_{k=1}^{L} b_k = 1\right)$, corresponding to a set of output class probabilities. During testing, search may occur if the baseline vigilance parameter $(\overline{\rho})$ is positive. Once a chosen F_2 node J meets the ART$_a$ matching criterion, the predicted outcome probability distribution is the $F_2 \to F^{ab}$ weight vector $(w_{J1} \ldots w_{Jk} \ldots w_{JL})$, normalized to 1 at F_0^b.

<u>(1) Variables</u>: $i = 1 \ldots 2M$, $j = 1 \ldots N$, $k = 1 \ldots L$

STM activation	LTM weights	$F_1 \to F_2$ signals	
x_i - F_1 (matching)	w_{ij} - $F_1 \leftrightarrow F_2$	S_j - Phasic	C - # committed nodes
y_j - F_2 (coding)	w_{jk} - $F_2 \to F^{ab}$	Θ_j - Tonic	ρ - ART$_a$ vigilance
z_k - F^{ab} (map field)		T_j - Total	

<u>(2) Signal rule</u>: Define the $F_1 \to F_2$ signal function $T_j = g\left(S_j, \Theta_j\right)$, where $g(0,0) = 0$ and $\dfrac{\partial g}{\partial S_j} > \dfrac{\partial g}{\partial \Theta_j} > 0$ for $S_j > 0$ and $\Theta_j > 0$.

E.g., $T_j = S_j + (1-\alpha)\Theta_j$ with $\alpha \in (0,1)$ (choice-by-difference) or

$T_j = S_j / \left(\alpha + 2M - \Theta_j\right)$ with $\alpha > 0$ (Weber law).

In ARTMAP, ART-EMAP, and ARTMAP-IC, the phasic signal component S_j equals $\sum_{i=1}^{2M} A_i \wedge w_{ij}$ and the tonic signal component Θ_j equals $\sum_{i=1}^{2M} \left(1 - w_{ij}\right)$.

<u>(3) Notation</u>

Minimum - $a \wedge b \equiv \min\{a, b\}$

(4) Parameters

Number of input components - $i = 1 \ldots 2M$

Number of coding nodes - $j = 1 \ldots N$

Number of output components - $k = 1 \ldots L$

Signal rule parameters - E.g., $\alpha \in (0,1)$ (choice-by-difference) or $\alpha > 0$ (Weber law)

Learning rate - $\beta \in [0,1]$, with $\beta = 1$ for fast learning

Baseline vigilance (ART_a) - $\bar{\rho} \in [0,1]$, with $\bar{\rho} = 0$ for maximal code compression

Map field vigilance - $\rho_{ab} \in [0,1]$, with $\rho_{ab} \cong 1$ for maximal output separation

Match tracking - ε, with $|\varepsilon|$ small.
MT+: $\varepsilon > 0$
MT-: $\varepsilon \le 0$
F_2 order constants - $0 < \Phi_N < \ldots < \Phi_j < \ldots < \Phi_1 < g(M,0)$, with all $\Phi_j \cong g(M,0)$.

(5) First iteration: $n = 1$

Input - $$A_i = \begin{cases} a_i^{(1)} & \text{if } 1 \le i \le M \\ 1 - a_i^{(1)} & \text{if } M+1 \le i \le 2M \end{cases}$$

Output - $b_k = b_k^{(1)} \quad k = 1 \ldots L$

$F_1 \leftrightarrow F_2$ weights - $w_{ij} = 1 \quad i = 1 \ldots 2M, \; j = 1 \ldots N$

$F_2 \to F^{ab}$ weights - $w_{jk} = 1 \quad j = 1 \ldots N, \quad k = 1 \ldots L$

Number of committed nodes - $C = 0$

Signal to uncommitted nodes - $T_j = \Phi_j \quad j = 1 \ldots N$

ART_a vigilance - $\rho = \bar{\rho}$

(6) Reset: New STM steady state at F_2 and F_1
Choose a category - Let J be the index of the F_2 node with maximal input T_j, i.e.,
$$T_J = \max\{T_1 \ldots T_N\}$$

Number of committed nodes - If $J > C$, set $C = J$

F_1 activation - $x_i = A_i \wedge w_{iJ} \quad i = 1 \ldots 2M$

(7) MTM: $F_1 \to F_2$ signal is refractory on the time scale of search

$T_J = 0$

(8) Reset or prediction: Check the F_1 matching criterion

If $\sum_{i=1}^{2M} x_i < \rho M$, go to (6) Reset

If $\sum_{i=1}^{2M} x_i \geq \rho M$, go to (9) Prediction

(9) Prediction:

F^{ab} activation - $z_k = b_k \wedge w_{Jk} \qquad k = 1 \ldots L$

(10) Match tracking or resonance: Check the F^{ab} matching criterion

If $\sum_{k=1}^{L} z_k < \rho_{ab}$, go to (11) Match tracking

If $\sum_{k=1}^{L} z_k \geq \rho_{ab}$, go to (12) Resonance

(11) Match tracking: Raise ρ to the point of ART$_a$ reset

$$\rho = \frac{1}{M} \sum_{i=1}^{2M} x_i + \varepsilon$$

Go to (6) Reset

(12) Resonance: New LTM weights on the time scale of learning

Old weights - $w_{iJ}^{old} = w_{iJ} \qquad i = 1 \ldots 2M, \qquad w_{Jk}^{old} = w_{Jk} \qquad k = 1 \ldots L$

Decrease $F_1 \leftrightarrow F_2$ weights - $w_{iJ} = (1-\beta) w_{iJ}^{old} + \beta\left(A_i \wedge w_{iJ}^{old}\right) \qquad i = 1 \ldots 2M$

Decrease $F_2 \rightarrow F^{ab}$ weights - $w_{Jk} = (1-\beta) w_{Jk}^{old} + \beta\left(b_k \wedge w_{Jk}^{old}\right) \qquad k = 1 \ldots L$

ART$_a$ vigilance recovery - $\rho = \overline{\rho}$

(13) Next iteration: Increase n by 1

New input - $A_i = \begin{cases} a_i^{(n)} & \text{if } 1 \leq i \leq M \\ 1 - a_i^{(n)} & \text{if } M+1 \leq i \leq 2M \end{cases}$

New output - $b_k = b_k^{(n)} \qquad k = 1 \ldots L$

New F_1 activation - $x_i = A_i \wedge w_{iJ} \qquad i = 1 \ldots 2M$

New $F_1 \rightarrow F_2$ signal to committed nodes

Phasic - $$S_j = \sum_{i=1}^{2M} A_i \wedge w_{ij} \qquad i = 1 \ldots 2M, \quad j = 1 \ldots C$$

Tonic - $$\Theta_j = \sum_{i=1}^{2M} \left(1 - w_{ij}\right) \qquad i = 1 \ldots 2M, \quad j = 1 \ldots C$$

Total - $$T_j = \begin{cases} g\left(S_j, \Theta_j\right) & j = 1 \ldots C, \quad j \neq J \quad ((2) \text{ Signal rule}) \\ 0 & j = J \quad \left(J^{th} \text{ node refractory}\right) \end{cases}$$

Go to (8) Reset or prediction

7.2 - Fuzzy ARTMAP Testing

During ARTMAP testing (Figure 6), $F_1 \leftrightarrow F_2$ categorization weights w_{ij} and $F_2 \rightarrow F^{ab}$ prediction weights w_{jk} are fixed. A test-set input **a** chooses an ART$_a$ category *J*, possibly following search, if $\bar{\rho} > 0$. Map field activation **z** then equals the $F_2 \rightarrow F^{ab}$ weight vector $\left(w_{J1} \ldots w_{Jk} \ldots w_{JL}\right)$, and the output vector **b** equals this vector normalized to 1. With fast learning, when **b** represents single output classes during training, only one component of **z** and **b** is positive, corresponding to a single class prediction. When **b** is distributed during training or learning is slow, **b** may represented a probability vector, distributed across output classes.

ARTMAP fast learning typically leads to different adaptive weights and recognition categories for different orderings of a given training set, even when the overall predictive accuracy of each such trained network is similar. The different category structures cause the location of test set inputs where errors occur to vary as the training set input orderings vary. A voting strategy uses several ARTMAP systems that are separately trained on one input set with different orderings. The final prediction for a given test set item is the one made by the largest number of networks in a voting "committee." Since the set of items making erroneous predictions varies from one ordering to the next, voting serves both to cancel many of the errors and to assign confidence estimates to competing predictions. A committee of about five voters has proved suitable in many examples, and the marginal benefits of voting are most apparent when the number of training samples is limited.

For voting, ARTMAP generates a set of prediction vectors for each of the trained networks produced by several different orderings of the training set inputs. The voting networks may average their output vectors **b** for each input **a**; or each voting network may choose one output class, with the predicted class being the one that receives the most votes.

(1) Test set input:

Input - $$A_i = \begin{cases} a_i & \text{if } 1 \leq i \leq M \\ 1 - a_i & \text{if } M + 1 \leq i \leq 2M \end{cases}$$

(2) $F_1 \rightarrow F_2$ signal:

Phasic - $$S_j = \sum_{i=1}^{2M} A_i \wedge w_{ij} \qquad i = 1 \ldots 2M, \quad j = 1 \ldots C$$

Tonic - $$\Theta_j = \sum_{i=1}^{2M} \left(1 - w_{ij}\right) \qquad i = 1 \ldots 2M, \quad j = 1 \ldots C$$

Total - $$T_j = \begin{cases} g\left(S_j, \Theta_j\right) & j = 1 \ldots C \quad \text{(Signal rule)} \\ \Phi_j & j = C + 1 \ldots N \end{cases}$$

(3) F_2 category choice:

- Let J be the index of the F_2 node with maximal signal T_j, i.e.,

$$T_J = \max\left\{T_1 \ldots T_N\right\}$$

(4) Output prediction:

$$b_k = \frac{w_{Jk}}{\sum_{\kappa=1}^{L} w_{J\kappa}} \qquad k = 1 \ldots L$$

8. AN ARTMAP PROTOTYPE APPLICATION: SATELLITE REMOTE SENSING

A new ARTMAP-based methodology for automatic mapping from Landsat Thematic Mapper (TM) and terrain data has been tested on a challenging remote sensing classification problem, using spectral and terrain features for vegetation classification in the Cleveland National Forest (Carpenter, Gjaja, Gopal, & Woodcock, 1995). After training at the pixel level, system capabilities are tested at the stand level, using sites not seen during training. ARTMAP performance was compared to those of maximum likelihood classifiers, as well as back propagation neural networks and K Nearest Neighbor (KNN) algorithms. ARTMAP learning, being fast, stable, and scalable, overcomes common limitations of back propagation, which did not give satisfactory performance on this problem. Best results were obtained using a hybrid system based on a convex combination of fuzzy ARTMAP and maximum likelihood predictions. The prototype remote sensing example below (Section 8.1) introduces each aspect of data processing and fuzzy ARTMAP classification (Section 8.2). The example shows how the network automatically constructs a minimal number of recognition categories to meet accuracy criteria (Section 8.3). A voting strategy (Section 8.4) improves prediction by training the system several times on different orderings of an input set. Voting assigns confidence estimates to competing predictions.

8.1 - A Prototype Remote Sensing Problem

Mapping vegetation from satellite remote sensing data has been an active area of research and development over a twenty year period (Hoffer et al., 1975; Strahler, Logan, & Bryant, 1978). Fuzzy ARTMAP has become the basis of a new systematic methodology for automatic classification of vegetation at the species level from multispectral and ancillary data.

A simplified remote sensing classification problem illustrates fuzzy ARTMAP dynamics. The prototype task is learning to identify one of three CALVEG (Matyas & Parker, 1980) vegetation classes (mixed conifer, coast live oak, southern mixed chaparral) for sites at which two spectral values

(Landsat TM1 and TM4) are known at each pixel. The prototype example is based on a data set collected at the Cleveland National Forest. Larger scale simulations on this data set predict 8 possible vegetation classes with inputs of up to 6 TM bands and 7 ancillary variables. In this more realistic setting, fuzzy ARTMAP performance compares favorably with that of maximum likelihood (Lillesand & Kiefer, 1994, pp. 594-596; Richards, 1993), K Nearest Neighbor (Duda & Hart, 1973), and back propagation (Rumelhart, Hinton, & Williams, 1986; Werbos, 1974). However, reducing the number of input dimensions to $M = 2$ (TM bands) and the number of output classes to $L = 3$ (vegetation classes) allows visual illustration of fuzzy ARTMAP dynamics, as follows.

The data set for the prototype remote sensing problem reports the vegetation class for each of 50 sites: 16 mixed conifer, 25 coast live oak, and 9 southern mixed chaparral (Table 1.A). The sites vary in size, averaging about 90 pixels each. Landsat spectral bands TM1 and TM4 constitute the data set input for each pixel, with values scaled to the interval $[0,1]$. Before training, 10 sites, representative of the vegetation class mix, are reserved as a test set. No pixels from these sites are used during training. The goal is to predict the correct vegetation class label for each of the 10 test set sites.

During training and testing, a given pixel corresponds to an ART_a input $\mathbf{a} \equiv (a_1, a_2)$, where a_1 is the value of TM1 and a_2 is the value of TM4 at that pixel. The corresponding ART_b input vector $\mathbf{b}$ represents the CALVEG vegetation class of the pixel's site:

$$\mathbf{b} = \begin{cases} (1,0,0) & \text{mixed conifer} \\ (0,1,0) & \text{coast live oak} \\ (0,0,1) & \text{southern mixed chaparral .} \end{cases} \qquad \text{(eq 26)}$$

During training, vector $\mathbf{b}$ informs the ARTMAP network of the vegetation class to which the pixel's site belongs. This supervised learning process allows adaptive weights to encode the correct association between $\mathbf{a}$ and $\mathbf{b}$. Simulations below examine the effect of training set size on predictive accuracy (Table 1.B). To generate a training set of a given size, pixels are selected at random from the entire training set, which represents approximately 3600 pixels in 40 sites. Other simulations show how voting can improve predictive accuracy (Table 1.C).

During testing, each test set pixel predicts a class, given the spectral band input values a_1 and a_2 for that pixel. Performance accuracy is measured both in terms of the percent of pixels that are correct and in terms of the fraction of sites that are correctly identified by a vote among pixels in the site.

The prototype remote sensing problem requires a trained network to predict the vegetation class (mixed conifer, coast live oak, or southern mixed chaparral) of a test set site, given TM bands 1 and 4 measured at each pixel in the site. This section illustrates fuzzy ARTMAP dynamics by showing how the network learns to make correct vegetation class predictions on this problem. Figure 7 illustrates why the problem is difficult: of the 4436 pixels in the data set (Table 1.A), many share spectral band values within and between the three vegetation classes, and the three classes are not linearly separable. In fact the problem proved to be too difficult for back propagation to make accurate predictions.

During the initial learning phase, pixels are selected one at a time, at random, from the 40 training set sites. Fuzzy ARTMAP is trained incrementally, with each TM band vector $\mathbf{a}$ presented just once. Following a search, if necessary, the network selects an ART_a category by activating an F_2^a node J for the input pixel, then learns to associate category J with the ART_b vegetation class K of the site in which the pixel is located. With fast learning, the class prediction K of each ART_a category J is permanent. If some input $\mathbf{a}$ with a different class prediction later selects this category, match tracking will raise ART_a vigilance ρ just enough to trigger a search for a different ART_a category. All prototype simulations use a Weber law signal rule (Section 7) with $\alpha \cong 0$ (conservative limit, Section 2), $\beta = 1$ (fast learning), and $\bar{\rho} = 0$ (forced choice prediction). The map field vigilance ρ_{ab} (Section 6) can have an arbitrary value between 0 and 1, since with fast learning and binary predictions the map field registers either a perfect match $(|\mathbf{z}| = 1)$ or a complete mismatch $(|\mathbf{z}| = 0)$.

8.2 - Geometry of Incremental Learning

Figure 8 illustrates fuzzy ARTMAP learning in response to the first 6 training set inputs, selected at random from the 40 training set sites. Input 1 (Figure 8a) represents a pixel that has a low TM1 value (a_1) and a high TM4 value (a_2) and that is found at a mixed conifer site (o). Input **a** selects the uncommitted F_2 node $J = 1$. During learning, all weights w_{Jk} from this node to the map field F^{ab} (Figure 3) decay to 0 except for the weight w_{JK} to the node K representing the correct vegetation class. Category $J = 1$ appears as the point box R_1

Input 2 (Figure 8b) also selects category $J = 1$. At the start of each input presentation, the ART$_a$ vigilance ρ equals the baseline vigilance $\bar{\rho}$, which here equals 0. Therefore, **a** meets the ART$_a$ matching criterion, so category $J = 1$ remains active and predicts, via the map field, that this new input is also from a mixed conifer site. Since this prediction is correct, field F^{ab} registers a perfect match and so meets the map field matching criterion. During learning the category box R_1 expands to include input point 2.

Input 3, from a coast live oak site (+), requires match tracking and search to learn the correct prediction as follows (Figure 8c). This input **a** first selects category $J = 1$. Again, since $\rho = \bar{\rho} = 0$, ART$_a$ accepts the new input into this category long enough to predict mixed conifer. However, the network now detects a predictive error, since the incorrect prediction sends the activity z_k of all map field nodes to 0. Match tracking increases ρ just enough to reset ART$_a$, where a new node $J = 2$ becomes active. Since uncommitted nodes meet the matching criterion for any ρ, node $J = 2$ remains active, establishing the point box R_2, which henceforth will predict coast live oak.

Input 4, again from a mixed conifer (o) site, shows how match tracking can create more than one box for each class. This feature allows ARTMAP to learn a set of decision rules of arbitrary complexity while minimizing predictive error. For example, concentric rings in an input space could be mapped to alternating category predictions. At the same time, setting $\bar{\rho}$ equal to 0 allows the network to maximize code compression, creating a new category only in response to a predictive error. Design principles that balance the two goals - minimum error, maximum compression - allow ARTMAP to learn correct predictions for a small category of rare cases embedded in a large category of common cases. Input 4 (Figure 8d) first selects the F_2 point category $J = 2$, which maximizes the choice function T_j (eq 2). Since this category predicts coast live oak, the map field registers a mismatch, which sends a match tracking signal to ART$_a$. This raises ρ until it is just above the match ratio $|\mathbf{A} \wedge \mathbf{w}_J||\mathbf{A}|^{-1}$, where $\mathbf{A} \equiv \left(\mathbf{a}, \mathbf{a}^c\right)$ is the complement coded input to F_1. The next category J that will be able to resonate, and so remain active long enough to make a class prediction, must now meet the stricter matching criterion imposed by the new, higher ART$_a$ vigilance ρ. Geometrically (Section 5), once node $J = 2$ leads to match tracking, a new active category J will now meet the ART$_a$ matching criterion only if the expanded box $R_J \oplus \mathbf{a}$ would be *smaller* than $R_2 \oplus \mathbf{a}$, where **a** is the current input. After match tracking, input 4 next selects category $J = 1$ (which actually would have made the correct prediction), but this category fails to meet the ART$_a$ matching criterion, since the box $R_1 \oplus \mathbf{a}$ would have been larger than $R_2 \oplus \mathbf{a}$. The input therefore activates the uncommitted node $J = 3$, which learns to predict mixed conifer.

Input 5 (Figure 8e) selects category node $J = 2$, which correctly predicts coast live oak (+), so no match tracking or ART_a search is invoked. During learning, as the weight vector $\mathbf{w}_2$ adapts according to equation (eq 11), the box R_2 expands to $R_2 \oplus \mathbf{a}$, where **a** represents the TM values of input 5. Since $\rho = \overline{\rho} = 0$, the size of $R_2 \oplus \mathbf{a}$ is unrestricted. Finally, input 6 (Figure 8f) selects and further expands box R_2. Weights remain unchanged during learning only if **a** is inside a selected box that has already learned to make the correct prediction. As training proceeds, category boxes cover more of the input space, so the case where weights remain unchanged during learning occurs increasingly often. If a finite input set can be presented repeatedly, all training set inputs learn to predict with 100% accuracy, provided that the set of input predictions is consistent, i.e., that no two identical inputs **a** make the same vegetation class prediction.

8.3 - Predictions of the Trained ARTMAP Network

As incremental learning proceeds, fuzzy ARTMAP creates a set of overlapping category boxes R_j, each predicting one of the three vegetation classes. By the time 100 training set pixel inputs have been selected at random from the 40 training set sites, fuzzy ARTMAP has created 8 categories (Table 1.B). Three of these categories predict mixed conifer, four predict coast live oak, and one predicts southern mixed chaparral. The 10 test set sites contain a total of 1108 pixels. After training on the first 100 inputs, network performance at this stage of learning was first measured by the number of correct vegetation class predictions the test set pixels were able to make. For each test set pixel, the TM band vector **a** selects one of the 8 ART_a categories, then predicts that its site belongs to the vegetation class associated with that category. After training on just 100 input points, 85.9% of the test set pixels correctly predicted the vegetation classes of their sites. A second performance measure examined the number of test set sites that would be correctly classified. This method counts the number of pixels in each site that predict each vegetation class, then selects the class chosen by the most pixels. At this stage of learning, having used only 3% of the training set pixels, 8 of the 10 test site vegetation classes were correctly identified. In this case, too few southern mixed chaparral exemplars had been presented for that class to easily win a majority at any site.

As the number of training set inputs increased, the pixel-level predictive accuracy increased only marginally, even decreasing as the number of training set inputs increased from 100 to 500 (Table 1.B). After presentation of all 3328 training set pixels, 89.3% of the test set pixels correctly predict the vegetation class of their site. However, site-level prediction improves steadily to 9/10 test set sites, after training on 500 inputs; and 10/10 sites, after training on 2000 inputs or on the full training set. This result highlights the observation that the pixel is often too small and noisy a unit to make an accurate prediction. However, a group of noisy pixel-level results can be pooled to form accurate mappings across functional regions or sites.

8.4 - Voting

A typical characteristic of fast learning is dependence of category structure upon the order of training set input presentation. For example, suppose that two fuzzy ARTMAP networks learn from a common input set that is presented in two different orders during training. The two networks might then each correctly predict 90% of the test set inputs, despite the fact that the two have significantly different internal input grouping rules, or category boxes, at ART_a. In particular, the test set inputs that the first network identifies correctly are typically different from those that the second network identifies correctly, despite the fact that both were trained on the same input set. ARTMAP voting uses this order dependence to advantage to improve and stabilize overall predictive performance, as follows.

Figure 9a-e illustrates the decision regions of the prototype remote sensing example after presentation of all 3328 training set inputs (Table 1.C). A decision region plot shows predictions all TM band inputs **a** would make if presented to the trained network. In Figure 7, data set points from mixed conifer sites were are represented by a circle (o), points from coast live oak sites by a plus (+), and points from southern mixed chaparral sites by a slash (/). The same marks indicate vegetation class predictions made by a network in response to spectral value inputs across the unit square. The rough decision boundaries in Figure 9a reflect the ambiguous predictions in the corresponding portion of the data set.

Figure 9a-e and Table 1.C show how network predictions can vary as a function of input order. Each of these five tests uses the same training set, presented in different, randomly chosen, orders. Decision boundaries vary, as do the number of ART_a categories (from 126 to 153), the number of correct test set pixels (from 84.8% to 89.4%), and the number of correct test set site identifications (from 8/10 to 10/10). Before knowing the test set answers, it would be difficult to decide which of these five networks would be the most accurate on novel data. ARTMAP voting chooses for each pixel the class prediction chosen by the largest number of the five "voting committee" networks. The size of each vote also provides a measure of confidence in each decisions. Confidence is typically lowest near decision boundaries. Figure 9f indicates how voting can smooth and stabilize decision boundaries. In addition, pixel-level performance on the voting network (91.0%) is better than that of any individual trained network, and site-level prediction is perfect (10/10).

9. ARTMAP VARIATIONS FOR APPLICATIONS

ART and ARTMAP networks feature winner-take-all (WTA) competitive coding, which groups inputs into disjoint recognition categories. Other neural network learning systems such as back propagation feature distributed coding, which can provide good noise tolerance and code compression but which typically requires slow learning. Fast learning tends to cause catastrophic forgetting in these networks, as it does in ART and ARTMAP networks in which the code representation is distributed. On the other hand, fast learning is often desirable for on-line adaptation to rapidly changing circumstances and for encoding of rare cases and large databases.

Variants of the basic ART and ARTMAP networks can acquire some of the advantages of distributed coding while maintaining fast learning capability. For example, ART-EMAP, which uses WTA codes for learning and distributed codes for testing (Section 9.1). Distributed prediction 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 (Section 9.3), which often feature inconsistent training input predictions, ARTMAP-IC (Carpenter & Markuzon, 1996) improves performance with a combination of distributed prediction, category instance counting, and a new match tracking search algorithm (Section 9.2). A voting strategy further improves prediction 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 problems such as category proliferation with noisy training sets, unless learning is slow. A new class of ART and ARTMAP networks permit fast distributed learning as well as performance (Section 9.4). These dART and dARTMAP systems (Carpenter, 1996) are now being analyzed and developed for future applications.

9.1 - ART-EMAP Distributed Prediction by the Q-max Rule

To improve performance in a noisy or ambiguous input environment, ART-EMAP adds spatial and temporal evidence accumulation processes to the basic ARTMAP system (Carpenter & Ross, 1993, 1995). ART-EMAP (Stage 1) distributes activity across category representations during performance. In a variety of studies, this device improves test-set predictive accuracy compared to ARTMAP, which is the same network with category choice during testing. Distributed test-set category activation also improves performance accuracy on medical database simulations, and further improvement is achieved by the addition of an instance counting measure (Section 9.2) that weights distributed predictions according to the number of training set inputs placed in each category.

ART-EMAP training is the same as ARTMAP training, with ART$_a$ category choice. During ART-EMAP testing, the degree of contrast enhancement at the competitive field F_2 is reduced, allowing distributed category activities y_j to form a combined prediction. The *Q-max rule* is a simple algorithm that approximates competitive contrast enhancement. The Q-max rule distributes F_2 activity y_j across the Q nodes that receive the largest $F_1 \rightarrow F_2$ inputs T_j, with y_j proportional to T_j. That is,

$$Q\text{-max rule:} \qquad y_j = \begin{cases} \dfrac{T_j}{\sum\limits_{\lambda \in \Lambda} T_\lambda} & \text{if } \lambda \in \Lambda \\ 0 & \text{if } \lambda \notin \Lambda, \end{cases} \qquad \text{(eq 27)}$$

where Λ is the set of Q nodes with the largest T_j values (Figure 10). The way a Q-max rule makes test set predictions is analogous to a K nearest neighbor (KNN) algorithm with $K = Q$. When $Q = 1$, the Q-max rule reduces to category choice. In the simulations below both ART-EMAP and ARTMAP-IC use the Q-max rule during testing.

Fair use of a Q-max rule requires *a priori* selection of Q, without knowledge of the test set exemplars. A general parameter selection method divides the original training set into a new training set and a complementary verification set, which can then be used to examine performance of the trained network for various parameters. Once parameters are selected by this method, the network can then start over, learning from the entire training set with the fixed set of parameters before making test set predictions. In choosing Q, the optimal value tends to scale with the size of the training set, so the optimal verification set value should be increased somewhat for testing. A second way to estimate Q is by a simple rule of thumb. ARTMAP, ART-EMAP, and ARTMAP-IC all employ the same training regime, using category choice. ART-EMAP and ARTMAP-IC then apply a Q-max rule during testing. Once a network is trained, the number (C) of committed F_2 category nodes is known, with each node having learned to predict one of the L possible output classes. On average, then, C / L category nodes predict each class. A reasonable *a priori* estimate sets Q equal to half that number, up to some maximum, say 30 category nodes. In other words:

$$\text{Rule-of-thumb } Q \text{ value:} \quad Q = \min\left\{\frac{C}{2L}, 30\right\}. \qquad \text{(eq 28)}$$

This estimate requires no separate verification step and has given good results on medical database simulations, where the number of output classes is often two, corresponding to good or bad outcomes. In the end, test set results can also be examined over a range of Q values to check for parameter sensitivity.

9.2 - Instance Counting

Instance counting biases distributed predictions according to the number of training set inputs classified by each F_2 node. Figure 10 illustrates how an ARTMAP network with an extra field F_3 can implement instance counting. During testing the $F_2 \rightarrow F_3$ input y_j is multiplied by the counting weight c_j to produce normalized F_3 activity Y_j, which projects to the map field F^{ab} for prediction (Figure 10). That is, for $j = 1,\ldots,N$, activity at the counting field F_3 is:

$$Y_j = \frac{c_j y_j}{\sum_{\eta=1}^{N} c_\eta y_\eta}. \qquad \text{(eq 29)}$$

The input U_k from F_3 to the k^{th} map field node is then:

$$U_k = \sum_{j=1}^{N} w_{jk} Y_j = \frac{\sum_{j=1}^{N} w_{jk} c_j y_j}{\sum_{j=1}^{N} c_j y_j} \qquad \text{(eq 30)}$$

for $k = 1,\ldots,L$. With choice at F_2,

$$Y_j = y_j = \begin{cases} 1 & \text{if } j = J \\ 0 & \text{if } j \neq J, \end{cases} \qquad \text{(eq 31)}$$

so $U_k = w_{Jk}$. With choice, map field activation and learning proceed as characterized in the training algorithm (Section 7.1).

The basic instance counting (IC) algorithm simply enumerates the training set inputs that activate each category, following search:

$$c_j^{(new)} = c_j^{(old)} + y_j, \qquad \text{(eq 32)}$$

with $c_j(0) = 0$. In the simulations below, c_j counts the number of times inputs select category j during training. Alternatives to this basic instance counting algorithm could be adapted to specific problems. One variation would train the entire network without instance counting, i.e., as a basic ARTMAP network; then calculate the counting weight vector $\mathbf{c}$ by re-presenting the training set, with either choice or Q-max distributed activation at F_2, and letting $\mathbf{c}$ enumerate the distributed activation vectors $\mathbf{y}$, summed across all training sets. With large training sets, it may also be useful to moderate the influence of some nodes that acquire an overwhelming number of training set instances. This could be accomplished by setting an upper bound on the c_j values or by having c_j grow logarithmically rather than linearly.

During testing (Section 7.2), when distributed F_2 activation is determined by a Q-max rule (eq 27):

$$U_k = \frac{\sum_{j=1}^{N} w_{jk} c_j y_j}{\sum_{j=1}^{N} c_j y_j} = \frac{\sum_{j \in \Lambda} w_{jk} c_j T_j}{\sum_{j \in \Lambda} c_j T_j}, \qquad \text{(eq 33)}$$

where Λ is the index set of the Q nodes with maximal $F_1 \rightarrow F_2$ input T_j. The net output probability distribution thus combines learned measures of pattern match (T_j), instance frequency (c_j), and class predictions (w_{jk}) for each category j.

9.3 - ARTMAP-IC Applied to a Medical Prediction Problem

The ARTMAP-IC neural network (Carpenter & Markuzon, 1996) adds distributed prediction and category instance counting to fuzzy ARTMAP. The ARTMAP match tracking algorithm, which controls search following a predictive error facilitates prediction with sparse or inconsistent data. Compared to the original match tracking algorithm MT+, MT- better approximates the network differential equations and compresses memory without loss of accuracy. Simulations below examine predictive accuracy on the Pima Indian diabetes medical databases. ARTMAP-IC networks results are favorable compared to those of logistic regression, K nearest neighbor (KNN), and the perceptron network ADAP, and also compared to the basic ARTMAP network and ART-EMAP. Voting, instance counting, and distributed representations combine to form confidence estimates for competing predictions.

The Pima Indian diabetes (PID) data set (Smith et al., 1988) was obtained from the UCI repository of machine learning databases (Murphy & Aha, 1992). The database task is to predict whether a patient will develop diabetes, based on eight clinical findings: age, the diabetes pedigree function, body mass, 2-hour serum insulin, triceps skin fold thickness, diastolic blood pressure, plasma glucose concentration, and number of pregnancies. Each patient represented in the database is a female of Pima Indian heritage who is at least 21 years old.

Smith et al. (1988) used the PID data set to evaluate ADAPtive learning routine (ADAP) (Smith, 1962), a type of perceptron (Rosenblatt, 1958, 1962). This study had 576 cases in the training set and 192 cases in the test set, and comparative simulations in this section keep the same training and test sets. About 39.9% of patients in the sample developed diabetes. Table 2 compares ADAP test set performance with that of logistic regression, KNN, and three ARTMAP networks. ARTMAP-IC uses the instance counting rule and the Q-max rule (eq 27) for distributed prediction. Comparative simulations show results for ART-EMAP (Stage 1), which is equivalent to ARTMAP-IC without instance counting; and for basic ARTMAP, which sets $Q = 1$ for category choice during testing. On average, the various ARTMAP networks $(\varepsilon = +0.0001)$, which share a common training regime, have 62 committed category nodes $(C = 62)$. With two output classes $(L = 2)$ the rule-of-thumb estimate (eq 28) for the size of distributed category representation sets $Q = 15$. Table 2 shows that ARTMAP-IC has the best test set performance, both in terms of the C-index and the number of correct test set predictions. MT- with $(\varepsilon = -0.0001)$ compresses memory even more, reducing the number of committed nodes from 62 to 45, with little deterioration in predictive accuracy. Compared to KNN, the ARTMAP networks compress memory by a factor of about 10:1.

9.4 - Distributed ART and Distributed ARTMAP

A new class of ART and ARTMAP models retain stable coding, recognition, and prediction, but allow arbitrarily distributed code representation during learning as well as performance (Carpenter, 1996). These 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. 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. The input signal T_j that activates the distributed code is a function of a *phasic component* S_j, which depends on the active input, and a *tonic component* Θ_j, which depends on prior learning but is independent of the current input, as in the fuzzy ARTMAP algorithm (Section 7.1). The computational properties of the phasic and tonic components are derived from a formal analysis of distributed pattern learning. However, these components can also be interpreted as postsynaptic membrane processes, with phasic terms mediated by ligand-gated receptors and tonic terms mediated by voltage-gated receptors (Nicholls, 1994). At each synapse, phasic and tonic terms balance one another and exhibit dual computational properties. For example, during learning with a constant input, phasic terms are constant while tonic terms may grow. Tonic components would then become larger for all inputs, but phasic components would become more selective, reducing the total coding signal sent by a significantly different input pattern. Dynamic weights that project to coding nodes obey a distributed instar leaning law and those that originate from coding nodes obey a distributed outstar learning law. Inputs activate distributed codes through phasic and tonic signal components with dual computational properties, and a parallel distributed match-reset-search process helps stabilize memory. When the code is winner-take-all, the unsupervised distributed ART model (dART) is computationally equivalent to fuzzy ART and the supervised distributed ARTMAP model (dARTMAP) is equivalent to fuzzy ARTMAP. With fast distributed learning, dART and dARTMAP networks are likely to further expand the domain of applications of the ART family of networks.

Acknowledgments - This research was supported in part by the National Science Foundation (NSF IRI 94-01659 and NSF SBR 93-00633) and the Office of Naval Research (ONR N00014-95-1-0409 and ONR N00014-95-0657).

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. **I**-512-517). Hillsdale, NJ: Lawrence Erlbaum Associates.

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

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. (1996). Distributed learning, recognition, and prediction by ART and ARTMAP neural networks. Submitted to *Neural Networks*. Technical Report CAS/CNS TR-96-004, Boston, MA: Boston University.

Carpenter, G.A., & Gjaja, M.N. (1994). Fuzzy ART choice functions. In *Proceedings of the World Congress on Neural Networks (WCNN'94)* (pp. **I**-713-722). Hillsdale, NJ: Lawrence Erlbaum Associates.

Carpenter, G.A., Gjaja, M.N., Gopal, S., & Woodcock, C. E. (1995). ART neural networks for remote sensing: Vegetation classification from Landsat TM and terrain data. Submitted to *IEEE Transactions on Geoscience and Remote Sensing*. Technical Report CAS/CNS TR-95-026, Boston, MA: Boston University.

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

Carpenter, G.A., & Grossberg, S. (1987b). ART 2: Self-organization of stable category recognition codes for analog input patterns. *Applied Optics*, **26**, 4919-4930.

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. (1991). 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. (1991). 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 Technical Report, 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. **III** - 649-656). Hillsdale, NJ: Lawrence Erlbaum Associates.

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* (pp. **II**-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. (1994a). Learning locomotion reflexes: A self-supervised neural system for a mobile robot. *Robotics and Autonomous Systems*, 12, 133-142.

Dubrawski, A., & Crowley, J.L. (1994b). Self-supervised neural system for reactive navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, San Diego, May, 1994 (pp. 2076 - 2081). Los Alamitos, CA: IEEE Computer Society Press.

Duda, R.O., & Hart, P.E. (1973). Pattern Classification and Scene Analysis. New York: Wiley.

Gan, K.W., & Lua, K.T. (1992). Chinese character classification using an Adaptive Resonance network. *Pattern Recognition*, **25**, 877-88.

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*. Lisbon, Portugal, DOSES, EUROSTAT. ECSC-EC-EAEC: Brussels, Luxembourg, pp. 55-68.

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

Grossberg, S. (1980). How does a brain build a cognitive code? *Psychological Review*, **87**, 1-51.

Ham, F.M., & Han, S.W. (1993). Quantitative study of the QRS complex using fuzzy ARTMAP and the MIT/BIH arrhythmia database. In *Proceedings of the World Congress on Neural Networks (WCNN'93)* (pp. **I**-207-211). Hillsdale, NJ: Lawrence Erlbaum Associates.

Hoffer, R.M., & Staff (1975). Natural resources mapping in mountainous terrain by computer analysis of ERTS-1 satellite data. Agricultural Experiment Station Research Bulletin 919, and LARS Contract Report 061575, W. Lafayette, IN: Purdue University, 124 pp.

Johnson, C. (1993). Agent learns user's behavior. *Electrical Engineering Times*, June 28, pp. 43, 46.

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

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.

Lillesand, T.M., & Kiefer, R.W. (1994). Remote Sensing and Image Interpretation. Third edition. New York: John Wiley.

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

Matyas, W.J., and Parker, I. (1980). CALVEG mosaic of existing vegetation of California. San Francisco: Regional Ecology Group, US Forest Service, Region 5, 630 Sansome Street. 27 pp.

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. **I**-228-232). Hillsdale, NJ: Lawrence Erlbaum Associates.

Moore, B. (1989). ART 1 and pattern clustering. In D. Touretzky, G. Hinton, & T. Sejnowski (Eds.), *Proceedings of the 1988 Connectionist Models Summer School* (pp. 174-185). San Mateo, CA: Morgan Kaufmann Publishers.

Murphy, P.M., & Aha, D.W. (1992). UCI repository of machine learning databases. Irvine, CA: University of California, Department of Information and Computer Science. [machine-readable data repository]

Murshed, N.A., Bortozzi, F., & Sabourin, R. (1995). Off-line signature verification, without *a priori* knowledge of class w_2. A new approach. In *Proceedings of ICDAR 95: The Third International Conference on Document Analysis and Recognition.*

Nicholls, D. G. (1994). Proteins, Transmitters and Synapses. Oxford: Blackwell Science Ltd.

Richards, J. (1993). Remote Sensing Digital Image Analysis: An Introduction (pp. 182-189). Springer-Verlag: Berlin.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, **65**, 386-408. Reprinted in J.A. Anderson & E. Rosenfeld (Eds.) (1988) Neurocomputing: Foundations of Research (pp. 18-27). Cambridge, MA: MIT Press.

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

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

Rumelhart, D.E., Hinton, G., & Williams, R. (1986). Learning internal representations by error propagation. In D.E. Rumelhart & J.L. McClelland (Eds.), Parallel Distributed Processing (pp. 318-362). Cambridge, MA: MIT Press.

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. **III**-661-664). Hillsdale, NJ: Lawrence Erlbaum Associates.

Smith, J.W. (1962). ADAP II, an adaptive routine for the LARC computer. Navy Management Office, Sept. 1962. (Available through the Logistics Management Institute Library.)

Smith, J.W., Everhart, J.E., Dickson, W.C., Knowler, W.C., & Johannes, R.S. (1988). Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In *Proceedings Symposium on Computer Applications and Medical Care* (pp. 261-265). IEEE Computer Society Press.

Strahler, A.H., Logan, T.L., & Bryant, N.A. (1978). Improving forest cover classification accuracy from Landsat by incorporating topographic information. *Proceedings of the 12th International Symposium on Remote Sensing of Environment* (pp. 927-942). Ann Arbor, MI: Environmental Research Institute of Michigan.

Suzuki, Y., Abe, Y., & Ono, K. (1993). Self-organizing QRS wave recognition system in ECG using ART 2. In *Proceedings of the World Congress on Neural Networks (WCNN'93)* (pp. **IV**-39-42). Hillsdale, NJ: Lawrence Erlbaum Associates.

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.

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.

Werbos, P. (1974). Beyond regression: New tools for prediction and analysis in the behavioral sciences. PhD Thesis, Cambridge, MA: Harvard University.

Wienke, D., & Kateman, G. (1994). Adaptive Resonance Theory based artificial neural networks for treatment of open-category problems in chemical pattern recognition - Application to UV-Vis and IR spectroscopy. *Chemometrics and Intelligent Laboratory Systems.*

Wienke, D., Xie, Y., & Hopke, P.K. (1994). An Adaptive Resonance Theory based artificial neural network (ART 2-A) for rapid identification of airborne particle shapes from their scanning electron microscopy images. *Chemometrics and Intelligent Laboratory Systems.*

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

Table 1: Prototype remote sensing simulations

A. Data set

Class label	# sites	# pixels
mixed conifer	16	1336
coast live oak	25	2752
southern mixed chaparral	9	348
TOTAL	50	4436

B. Fuzzy ARTMAP Incremental Learning

Training set (# pixels)	Categories (# F_2^a nodes)	Test set pixels (% correct)	Test set sites (# correct)
100	8	85.9%	8/10
500	21	83.2%	9/10
2000	72	88.5%	10/10
3328	126	89.3%	10/10

C. Voting

Input ordering (Figure 9)	Categories (# F_2^a nodes)	Test set pixels (% correct)	Test set sites (# correct)
(a)	126	89.3%	10/10
(b)	131	86.8%	9/10
(c)	139	86.8%	9/10
(d)	153	89.4%	9/10
(e)	133	84.8%	8/10
average	136	87.4%	9/10
voting	---	91.0%	10/10

Table 2: Pima Indian diabetes (PID) simulations

Model	**Correct predictions**	**C-index**	**Compression factor**
logistic regression	77%	0.84	--
ADAP	76%	--	--
ARTMAP (*K*=1) [MT+: ε = +0.0001]	66%	0.76	9.3

	Q = 15	12≤Q ≤19	Peak %	[C-index. Q]	Compression
KNN	77%	76-77%	77%	[0.80, Q=13-15]	1
ART-EMAP [MT+: ε = +0.0001]	76%	76-78%	78%	[0.87, Q=13]	9.3
ARTMAP-IC [MT+: ε = +0.0001]	79%	79-80%	80%	[0.87, Q=9-13]	9.3
	Q = 15	13≤Q ≤17			
ARTMAP-IC [MT-: ε = -0.0001]	81%	80-81%	81%	[0.88, Q=15]	9.3
	Q = 11	8≤Q ≤14			
ARTMAP-IC [MT-: ε = -0.01]	79%	78-81%	81%	[0.87, Q=9]	12.8

FIGURE CAPTIONS

Figure 1: In ART 1 and fuzzy ART, the F_2 the node $(j = J)$ that receives the largest input T_j from F_1 becomes active. Activity $\mathbf{x}$ at the field F_1 reflects the match between the bottom-up input $\mathbf{I}$ and the top-down input, which is equal to the weight vector $\mathbf{w}_J$. When $\mathbf{x}$ 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.

Figure 2. ART search for an F_2 code. (a) The input vector $\mathbf{A}$ generates the F_1 activity vector $\mathbf{x}$ as it activates the orienting subsystem Ω. Activity $\mathbf{x}$ both inhibits Ω and generates an $F_1 \rightarrow F_2$ signal. A bottom-up adaptive filter transforms $\mathbf{x}$ into the F_2 input vector $\mathbf{T}$, which activates the STM pattern $\mathbf{y}$ across F_2. (b) A top-down adaptive filter transforms $\mathbf{y}$ into the category representation vector $\mathbf{V}$. Where $\mathbf{V}$ mismatches $\mathbf{A}$, F_1 registers a diminished STM activity pattern $\mathbf{x}^*$. The resulting reduction of total STM reduces the total inhibitory signal from F_1 to Ω. (c) If the ART matching criterion fails, Ω releases a nonspecific signal that resets the STM pattern $\mathbf{y}$ at F_2. (d) Since reset inhibits $\mathbf{y}$, it also eliminates the top-down signal $\mathbf{V}$, so $\mathbf{x}$ can be reinstated at F_1. However, enduring traces of the prior reset allow $\mathbf{x}$ to activate a different STM pattern $\mathbf{y}^*$ at F_2. If the top-down signal due to $\mathbf{y}^*$ also mismatches $\mathbf{A}$ at F_1, then the search for an F_2 code that satisfies the matching criterion continues. (Carpenter & Grossberg, 1987a)

Figure 3. Fuzzy ART embedded in a simplified ARTMAP network. In the fuzzy ART algorithm, $\mathbf{w}_j$ denotes both the bottom-up weight vector and the top-down weight vector, with $w_{ij} = w_{ji}$. The ARTMAP network computes classification probabilities, with $|\mathbf{b}| = 1$ at an output field F_0^b.

Figure 4. Fuzzy ART category boxes, with $M = 2$. (a) In complement coding form, each weight vector $\mathbf{w}_j$ has a geometric interpretation as a rectangle R_j with corners $(\mathbf{u}_j, \mathbf{v}_j)$. (b) During fast learning, R_J expands to $R_J \oplus \mathbf{a}$, the smallest rectangle that includes R_J and $\mathbf{a}$, provided that $|R_J \oplus \mathbf{a}| \leq 2(1-\rho)$. (c) With fuzzy ART fast learning and complement coding, the j^{th} category rectangle R_j includes all those vectors $\mathbf{a}$ in the unit square that have activated category j without reset. The weight vector $\mathbf{w}_j$ equals $\left(\wedge_j \mathbf{a}, (\vee_j \mathbf{a})^c\right)$.

Figure 5. ARTMAP architecture. The ART$_a$ complement coding preprocessor transforms the M_a-vector $\mathbf{a}$ into the $2M_a$-vector $\mathbf{A}=(\mathbf{a},\mathbf{a}^c)$ at the ART$_a$ field F_0^a. $\mathbf{A}$ is the input vector to the ART$_a$ field F_1^a. Similarly, the input to F_1^b is the $2M_b$-vector $\mathbf{B}=(\mathbf{b},\mathbf{b}^c)$. When ART_b disconfirms a prediction of ART$_a$, map field inhibition induces the match tracking process. Match tracking raises the ART$_a$ vigilance ρ_a to just above the F_1^a-to-F_0^a match ratio $\left|\mathbf{x}^a\right| / |\mathbf{A}|$. This triggers an ART$_a$ search which leads either to an ART$_a$ category that correctly predicts $\mathbf{b}$ or to a previously uncommitted ART$_a$ category node. (Carpenter, Grossberg, & Reynolds, 1991)

Figure 6. During ARTMAP testing, an input $\mathbf{a}$ activates the J^{th} F_2 category node. The map field weights w_{Jk} then to form a prediction vector $\mathbf{z}$, which may be distributed. The network computes classification probabilities, with $|\mathbf{b}| = 1$, at the output field F_0^b.

Figure 7: Prototype remote sensing inputs. Each point shows the scaled Landsat spectral band components a_1 (TM1 - blue) and a_2 (TM4 - near infrared) of the ART$_a$ input vector **a**. Points o are found in mixed conifer sites, points + are found in coast live oak sites, and points / are found in southern mixed chaparral sites. Data set values are taken from the Cleveland National Forest.

Figure 8: Prototype remote sensing example: Fuzzy ARTMAP incremental learning in response to the first 6 training set points. Inputs 1(a), 2(b), and 4(d) are from mixed conifer sites (o) and inputs 3(c), 5(e), and 6(f) are from coast live oak sites (+). After learning, inputs 1 and 2 have established the ART$_a$ category $J=1$, which maps to mixed conifer; inputs 3, 5, and 6 have established category $J=2$, which maps to coast live oak; and input 4 has established the point category $J=3$, which maps to mixed conifer. Southern mixed chaparral, with sites that include less than 8% of the pixels, happened not to be represented among the first 6 inputs, which were selected at random.

Figure 9: Prototype remote sensing example: Fuzzy ARTMAP voting. (a)-(e) Fuzzy ARTMAP networks trained on a common set of 3328 inputs presented in five different, random orders show variations in decision region geometry. Points marked by a circle (o) predict mixed conifer, points marked by a plus (+) predict coast live oak, and points marked by a slash (/) predict southern mixed chaparral. Pixel-level predictive accuracy ranges from 84.8% (e) to 89.4% (d) while site-level predictive accuracy ranges from 8/10 (e) to 10/10 (a) (Table 1.C). (f) Voting across the five trained networks boosts pixel-level accuracy to 91.0% and site-level accuracy to 10/10. Blank spaces indicate a 2-2-1 tie among the voters.

Figure 10. During testing, an input activates Q category nodes, in proportion to the input from F_1 to the category field F_2. After multiplication by the instance counting weights to produce distributed activation Y_j at F_3, the Q active nodes project to the map field F^{ab} via the map field weights w_{jk} to form a distributed prediction vector $\mathbf{U}$. The network then computes classification probabilities, with $|\mathbf{b}|=1$ at an output field F_0^b.

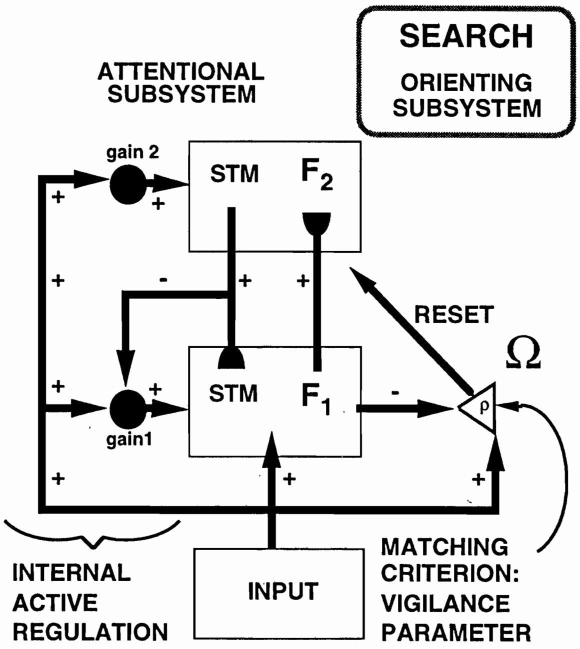

Figure 1: In ART 1 and fuzzy ART, the F_2 the node $(j = J)$ that receives the largest input T_j from F_1 becomes active. Activity **x** at the field F_1 reflects the match between the bottom-up input **I** and the top-down input, which is equal to the weight vector $\mathbf{w}_J$. When **x** fails to 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.

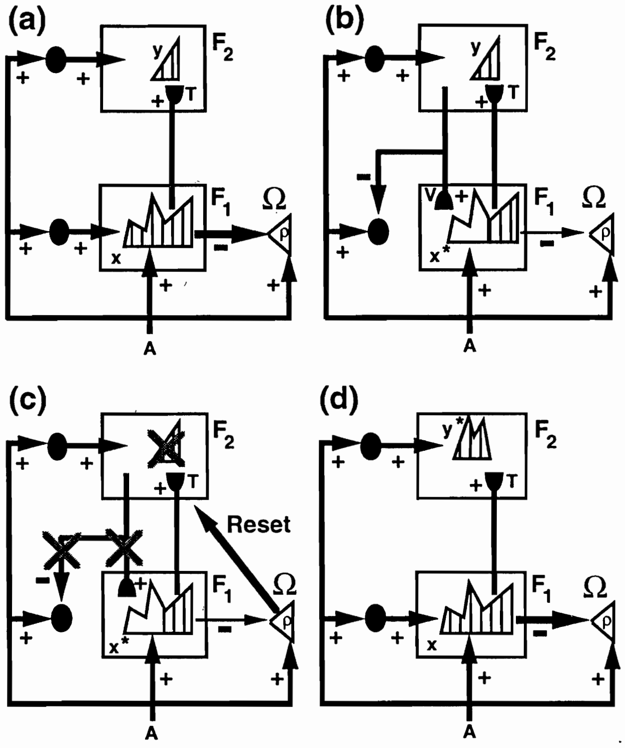

Figure 2. ART search for an F_2 code. (a) The input vector A generates the F_1 activity vector x as it activates the orienting subsystem Ω. Activity x both inhibits Ω and generates an $F_1 \rightarrow F_2$ signal. A bottom-up adaptive filter transforms x into the F_2 input vector **T**, which activates the STM pattern y across F_2. (b) A top-down adaptive filter transforms y into the category representation vector **V**. Where **V** mismatches A, F_1 registers a diminished STM activity pattern x^*. The resulting reduction of total STM reduces the total inhibitory signal from F_1 to Ω. (c) If the ART matching criterion fails, Ω releases a nonspecific signal that resets the STM pattern y at F_2. (d) Since reset inhibits y, it also eliminates the top-down signal V, so x can be reinstated at F_1. However, enduring traces of the prior reset allow x to activate a different STM pattern y^* at F_2. If the top-down signal due to y^* also mismatches A at F_1, then the search for an F_2 code that satisfies the matching criterion continues. (Carpenter and Grossberg, 1987a)

ARTMAP: Training

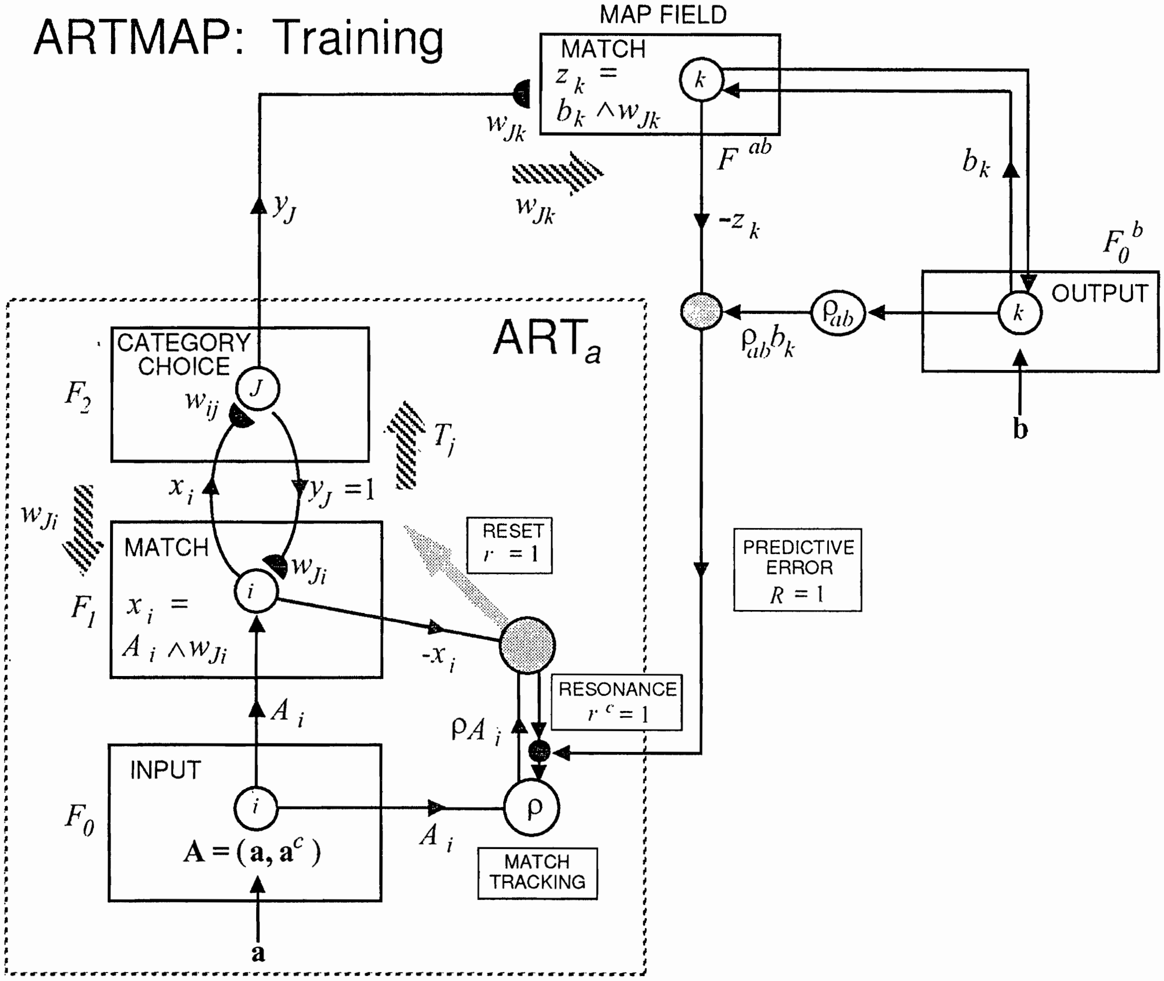

Figure 3. Fuzzy ART embedded in a simplified ARTMAP network. In the fuzzy ART algorithm, $\mathbf{w}_j$ denotes both the bottom-up weight vector and the top-down weight vector, with $w_{ij} = w_{ji}$. The ARTMAP network computes classification probabilities, with $|\mathbf{b}| = 1$ at an output field F_0^b.

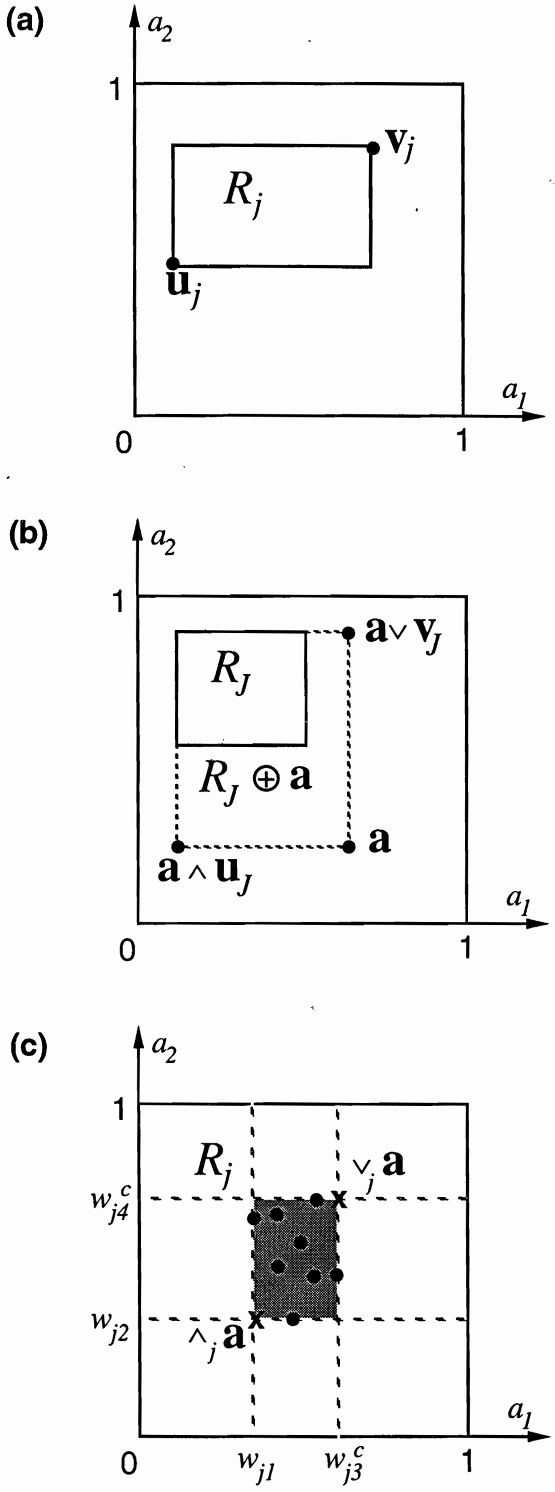

Figure 4. Fuzzy ART category boxes, with $M = 2$. (a) In complement coding form, each weight vector $\mathbf{w}_j$ has a geometric interpretation as a rectangle R_j with corners $(\mathbf{u}_j, \mathbf{v}_j)$. (b) During fast learning, R_J expands to $R_J \oplus \mathbf{a}$, the smallest rectangle that includes R_J and $\mathbf{a}$, provided that $|R_J \oplus \mathbf{a}| \le 2(1-\rho)$. (c) With fuzzy ART fast learning and complement coding, the j^{th} category rectangle R_j includes all those vectors $\mathbf{a}$ in the unit square that have activated category j without reset. The weight vector $\mathbf{w}_j$ equals $\left(\wedge_j \mathbf{a}, \left(\vee_j \mathbf{a}\right)^c\right)$.

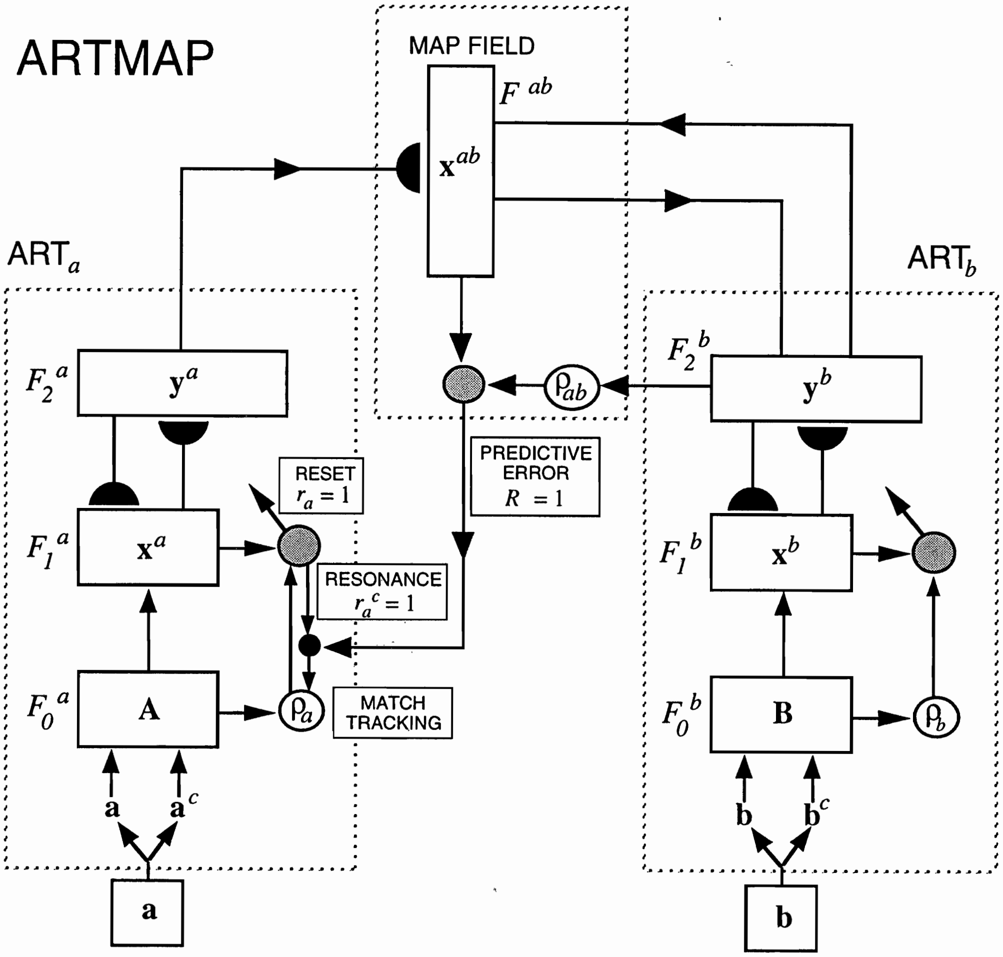

Figure 5. ARTMAP architecture. The ART$_a$ complement coding preprocessor transforms the M_a-vector **a** into the $2M_a$-vector $\mathbf{A}=(\mathbf{a},\mathbf{a}^c)$ at the ART$_a$ field F_0^a. **A** is the input vector to the ART$_a$ field F_1^a. Similarly, the input to F_1^b is the $2M_b$-vector $\mathbf{B}=(\mathbf{b},\mathbf{b}^c)$. When ART_b disconfirms a prediction of ART$_a$, map field inhibition induces the match tracking process. Match tracking raises the ART$_a$ vigilance ρ_a to just above the F_1^a-to-F_0^a match ratio $|\mathbf{x}^a|/|\mathbf{A}|$. This triggers an ART$_a$ search which leads either to an ART$_a$ category that correctly predicts **b** or to a previously uncommitted ART$_a$ category node. (Carpenter, Grossberg, and Reynolds, 1991)

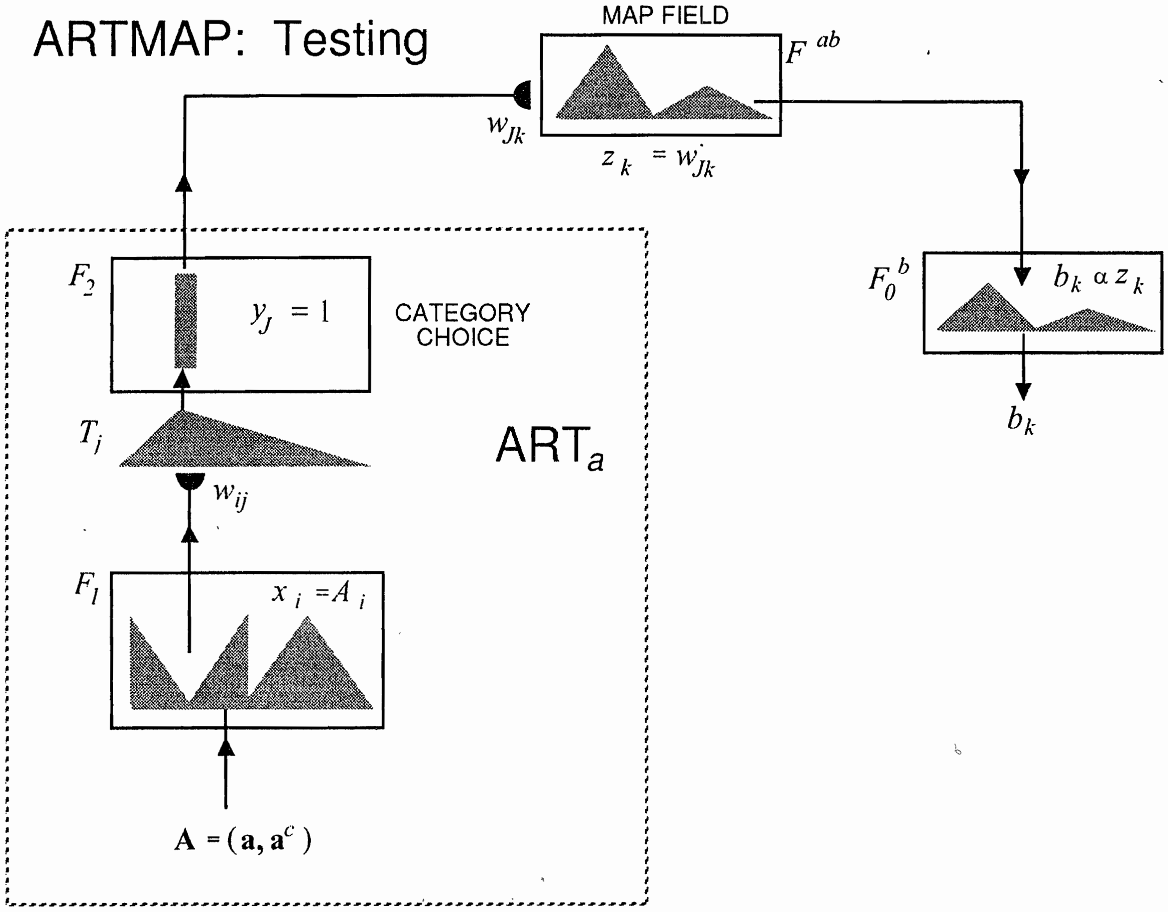

Figure 6. During ARTMAP testing, an input **a** activates the J^{th} F_2 category node. The map field weights w_{Jk} then to form a prediction vector **z**, which may be distributed. The network computes classification probabilities, with $|\mathbf{b}| = 1$, at the output field F_0^b.

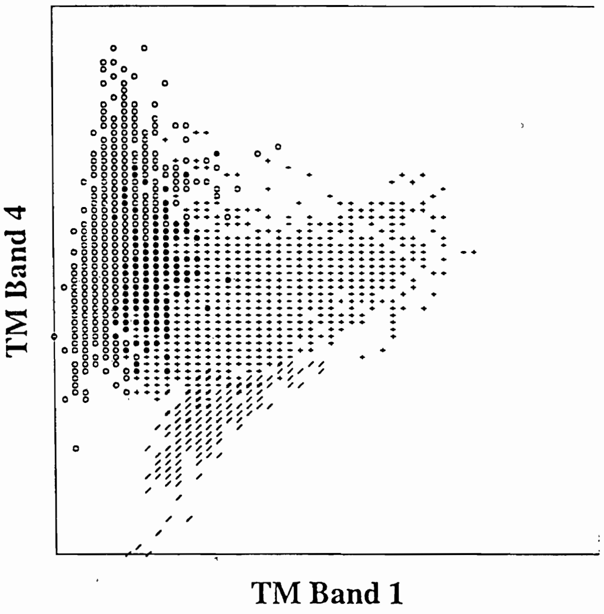

Figure 7: Prototype remote sensing inputs. Each point shows the scaled Landsat spectral band components a_1 (TM1 - blue) and a_2 (TM4 - near infrared) of the ART$_a$ input vector **a**. Points o are found in mixed conifer sites, points + are found in coast live oak sites, and points / are found in southern mixed chaparral sites. Data set values are taken from the Cleveland National Forest.

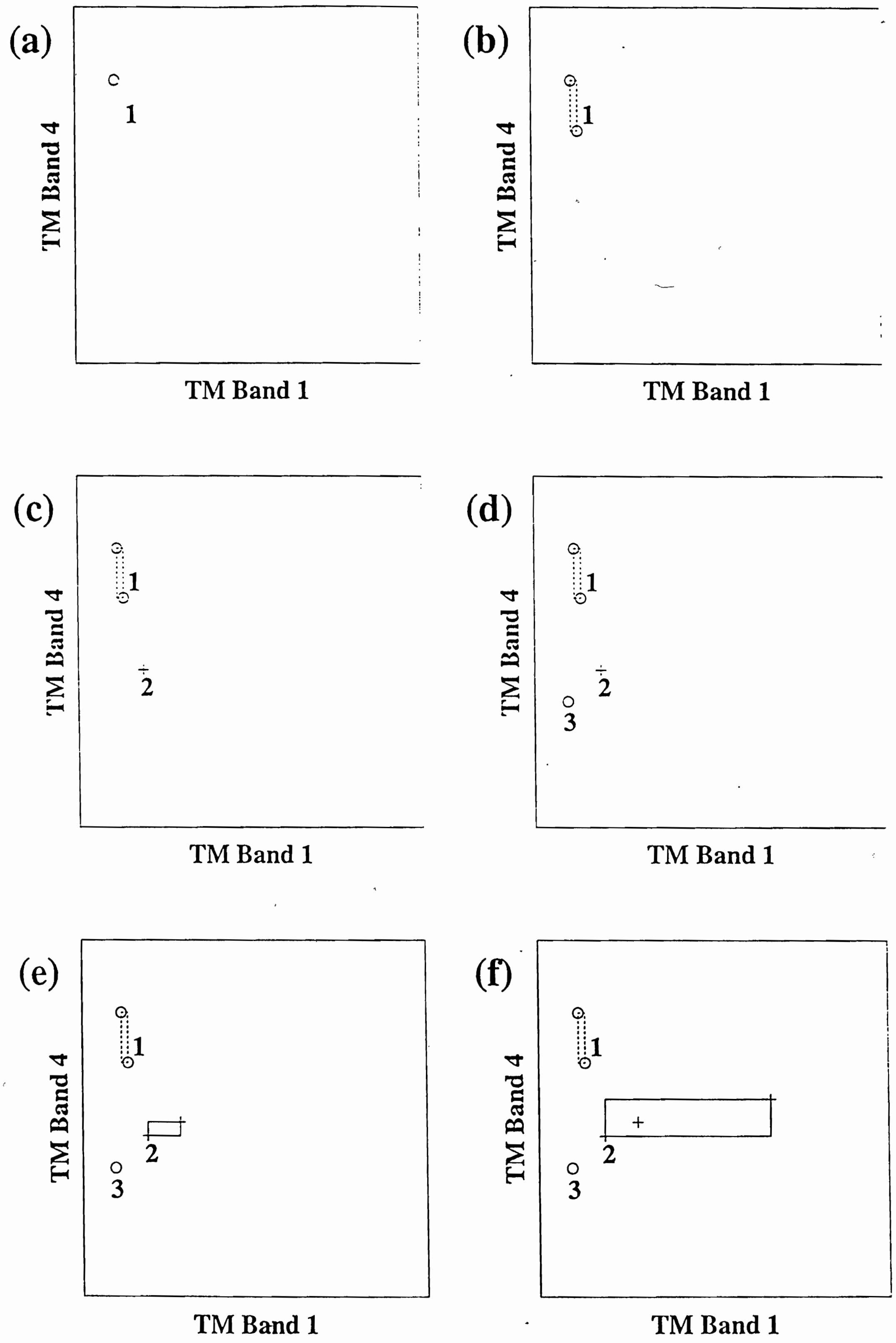

Figure 8: Prototype remote sensing example: Fuzzy ARTMAP incremental learning in response to the first 6 training set points. Inputs 1(a), 2(b), and 4(d) are from mixed conifer sites (o) and inputs 3(c), 5(e), and 6(f) are from coast live oak sites (+). After learning, inputs 1 and 2 have established the ART$_a$ category $J = 1$, which maps to mixed conifer; inputs 3, 5, and 6 have established category $J = 2$, which maps to coast live oak; and input 4 has established the point category $J = 3$, which maps to mixed conifer. Southern mixed chaparral, with sites that include less than 8% of the pixels, happened not to be represented among the first 6 inputs, which were selected at random.

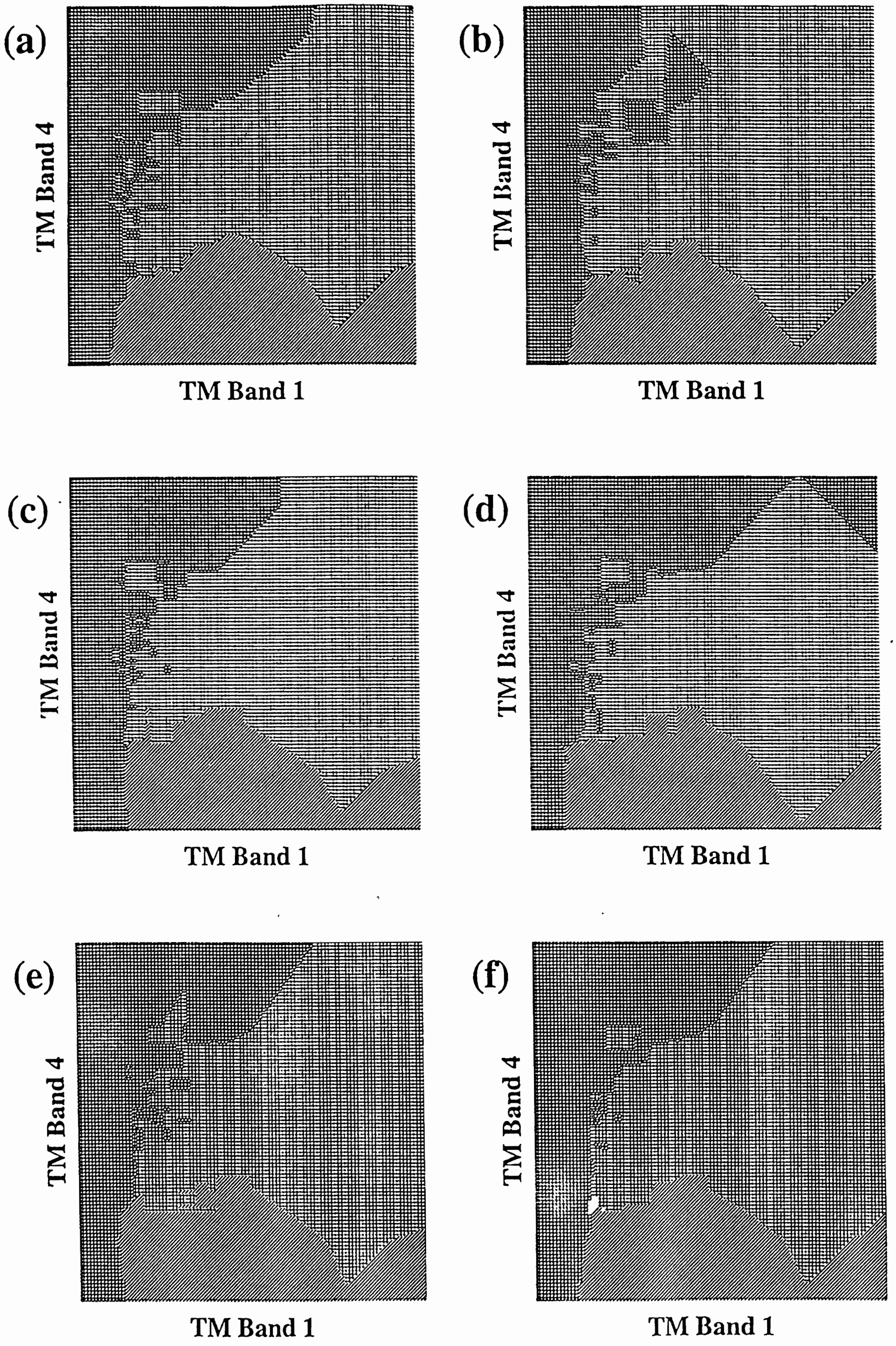

Figure 9: Prototype remote sensing example: Fuzzy ARTMAP voting. (a)-(e) Fuzzy ARTMAP networks trained on a common set of 3328 inputs presented in five different, random orders show variations in decision region geometry. Points marked by a circle (o) predict mixed conifer, points marked by a plus (+) predict coast live oak, and points marked by a slash (/) predict southern mixed chaparral. Pixel-level predictive accuracy ranges from 84.8% (e) to 89.4% (d) while site-level predictive accuracy ranges from 8/10 (e) to 10/10 (a) (Table 1.C). (f) Voting across the five trained networks boosts pixel-level accuracy to 91.0% and site-level accuracy to 10/10. Blank spaces indicate a 2-2-1 tie among the voters.

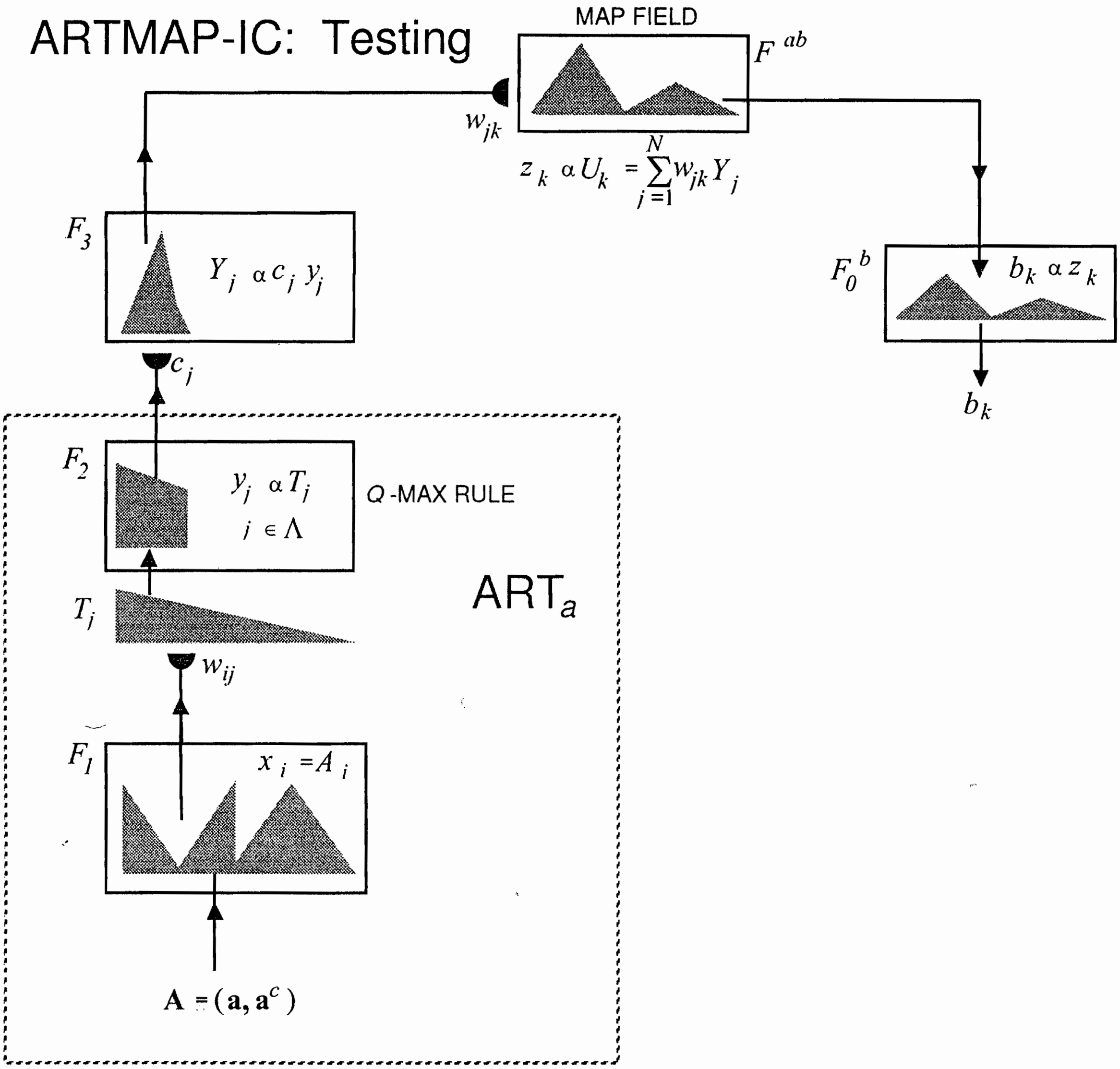

Figure 10. During testing, an input activates Q category nodes, in proportion to the input from F_1 to the category field F_2. After multiplication by the instance counting weights to produce distributed activation Y_j at F_3, the Q active nodes project to the map field F^{ab} via the map field weights w_{jk} to form a distributed prediction vector $\mathbf{U}$. The network then computes classification probabilities, with $|\mathbf{b}| = 1$ at an output field F_0^b.