FUZZY ART

Gail A. Carpenter

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

November, 1993

In: Bart Kosko
Fuzzy Engineering
Carmel, IN: Prentice Hall, 1994

Technical Report CAS/CNS-TR-93-059
Boston, MA: Boston University

This research was supported in part by ARPA (ONR N00014-92-J-4015), the National Science Foundation (NSF IRI 90-00530), and the Office of Naval Research (ONR N00014-91-J-4100).

Acknowledgements: The author wishes to thank Cynthia E. Bradford and Diana J. Meyers for their valuable assistance in the preparation of the manuscript.

ABSTRACT

Adaptive Resonance Theory (ART) models are real-time neural networks for category learning, pattern recognition, and prediction. Unsupervised fuzzy ART and supervised fuzzy ARTMAP synthesize fuzzy logic and ART networks by exploiting the formal similarity between the computations of fuzzy subsethood and the dynamics of ART category choice, search, and learning. Fuzzy ART self-organizes stable recognition categories in response to arbitrary sequences of analog or binary input patterns. It generalizes the binary ART 1 model, replacing the set-theoretic intersection ($\cap$) with the fuzzy intersection ($\wedge$), or component-wise minimum. A normalization procedure called complement coding leads to a symmetric theory in which the fuzzy intersection and the fuzzy union ($\vee$), or component-wise maximum, play complementary roles. Complement coding preserves individual feature amplitudes while normalizing the input vector, and prevents a potential category proliferation problem. Adaptive weights start equal to one and can only decrease in time. A geometric interpretation of fuzzy ART represents each category as a box that increases in size as weights decrease. A matching criterion controls search, determining how close an input and a learned representation must be for a category to accept the input as a new exemplar. A vigilance parameter (ρ) sets the matching criterion and determines how finely or coarsely an ART system will partition inputs. High vigilance creates fine categories, represented by small boxes. Learning stops when boxes cover the input space. With fast learning, fixed vigilance, and an arbitrary input set, learning stabilizes after just one presentation of each input. A fast-commit slow-recode option allows rapid learning of rare events yet buffers memories against recoding by noisy inputs.

Fuzzy ARTMAP unites two fuzzy ART networks to solve supervised learning and prediction problems. A Minimax Learning Rule controls ARTMAP category structure, conjointly minimizing predictive error and maximizing code compression. Low vigilance maximizes compression but may therefore cause very different inputs to make the same prediction. When this coarse grouping strategy causes a predictive error, an internal match tracking control process increases vigilance just enough to correct the error. ARTMAP automatically constructs a minimal number of recognition categories, or "hidden units," to meet accuracy criteria. An ARTMAP voting strategy improves prediction by training the system several times using different orderings of the input set. Voting assigns confidence estimates to competing predictions given small, noisy, or incomplete training sets. ARPA benchmark simulations illustrate fuzzy ARTMAP dynamics. The chapter also compares fuzzy ARTMAP to Salzberg's Nested Generalized Exemplar (NGE) and to Simpson's Fuzzy Min-Max Classifier (FMMC); and concludes with a summary of ART and ARTMAP applications.

Match-Based Learning and Error-Based Learning

When is a dog not a dog?: A stable learning system incorporates crucial new data into an existing memory system without destroying old memories. We effortlessly remember the a dog is still a dog, even as we learn that this dog is a Dalmatian. In a complex world, new information often contradicts the old, while both are important and correct.

An ART network constructs new memories based on the success or failure of old memories, as they guide the system in the world. Some categories are coarse (dog) or fine (Dalmatian), as needed. When I expect to hear "dog" but the answer is "Dalmatian," I am startled into paying attention to features that I had previously ignored. When I learn to recognize a Dalmatian, I do not forget about dogs as a group. Similarly, ART memories encode attended features, rather than the entire set of features that happen to be present at the moment. This is the basis for the network's stability.

Match-based learning and stable coding: ART memories are stable in a complex world because the learning process is *match-based*. Memories change only when attended portions of the external world match our internal expectations—or when something completely new occurs. When the external world fails to match an ART network's expectations or predictions, a search process activates a new category. The new category represents a new hypothesis about what is important in the present environment. Match-based learning is the defining characteristic of ART networks.

Boeing Neural Information Retrieval System: Code stability is one of the main reasons ART networks are selected for applications. One example of such a technology transfer is the Boeing Neural Information Retrieval System (NIRS) (Caudell, 1993; Smith, Escobedo, and Caudell, 1993), in which ART networks are the critical system components. NIRS encodes a parts inventory in the form of 2-D and 3-D drawings. The system creates a compressed but stable memory structure for later retrieval by design engineers. The resulting neural database reduces inventory size by a factor of nine, thus alleviating a severe memory proliferation problem and permittting efficient reuse of stored designs. NIRS has moved from beta testing to implementation in CAD systems for design, of the Boeing 777, and manufacturing, of the Boeing 747 and 767 planes.

Error-based learning: Match-based learning generates a stable recognition code in a large, complex, evolving environment. A match-based learning system is thus well suited to problems such as the Boeing CAD neural database, which creates its own expert system as a function of experience. However, qualitatively different types of learning problems also exist. For example, as we grow, our eyes and limbs need to learn, or adapt, to their own internal changes so that we can pick up a pencil as an adult as well as we could at age two. As adults we have no need for the sensory-motor maps that we learned as babies. These codes need not, therefore, be stable in the sense that knowledge systems, such as language, need to be stable. Layers of old motor maps would most likely be a great nuisance.

Neural networks that employ *error-based* learning are well suited to adaptive problems. Error-based learning systems include the perceptron (Rosenblatt, 1958, 1962) and multi-layer perceptrons such as back propagation (Werbos, 1974; Rumelhart, Hinton, and Williams, 1986). In these systems, an error causes memories to change so that the same input, seen again, would give an answer that was closer to the "correct" one. If I see a dog and know it is a dog, but am then told it is a Dalmatian, a serious error has occurred. An error-based

network will shift the weights in such a way that the next response will be toward Dalmatian, away from dog. If this happens several times in a row, I will learn to respond "Dalmatian," but will completely forget that a dog is still a dog. Error-based learning is, hereby, subject to "catastrophic forgetting." This kind of forgetting is desirable, however, if the error signal tells me that I have reached too far to touch the pencil.

ART and Fuzzy Logic

Stephen Grossberg (1976) introduced Adaptive Resonance Theory (ART) as a theory of human cognitive information processing. The theory has led to an evolving series of real-time neural network models for unsupervised and supervised category learning and pattern recognition. These models form stable recognition categories in response to arbitrary input sequences with either fast or slow learning. Unsupervised ART networks include ART 1 (Carpenter and Grossberg, 1987a), which stably learns to categorize binary input patterns presented in an arbitrary order; ART 2 (Carpenter and Grossberg, 1987b), which stably learns to categorize either analog or binary input patterns presented in an arbitrary order; and ART 3 (Carpenter and Grossberg, 1990), which carries out parallel search, or hypothesis testing, of distributed recognition codes in a multi-level network hierarchy. Many of the ART papers are collected in the anthology **Pattern Recognition by Self-Organizing Neural Networks** (Carpenter and Grossberg, 1991).

A supervised network architecture, called ARTMAP, self-organizes categorical mappings between m-dimensional input vectors and n-dimensional output vectors. ARTMAP's internal 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. Fuzzy ART (Carpenter, Grossberg, and Rosen, 1991a) generalizes ART 1 to learn stable recognition categories in response to analog and binary input patterns. When fuzzy ART replaces ART 1 in an ARTMAP system, the resulting fuzzy ARTMAP architecture (Carpenter, Grossberg, Markuzon, Reynolds, and Rosen, 1992) rapidly learns stable categorical mappings between analog or binary input and output vectors. Fuzzy ARTMAP learns to classify inputs by a fuzzy set of features, or a pattern of fuzzy membership values between 0 and 1, that indicate the extent to which each feature is present. Where set-theoretic operations describe ART 1 dynamics fuzzy set-theoretic operations (Kosko, 1986; Zadeh, 1965) describe fuzzy ART dynamics.

Simulations in this chapter illustrate fuzzy ARTMAP performance. Simulation inputs are 2-D analog patterns that are not necessarily interpreted as fuzzy sets, but that illustrate the properties of the system. 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 simulation is similar. The different category structures cause the set of test set items where errors occur to vary from one simulation to the next. A *voting strategy* uses an ARTMAP system that is trained several times 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 simulations. Since the set of items making erroneous predictions varies from one simulation to the next, voting both cancels many of the errors and assigns confidence estimates to competing predictions. The chapter concludes with some fuzzy ART and ARTMAP applications.

ART Dynamics

Fuzzy ART incorporates the basic features of all ART systems, notably, pattern matching between bottom-up input and top-down learned prototype vectors. This matching process leads either to a resonant state that focuses attention and triggers stable prototype learning or to a self-regulating parallel memory search. If the search ends with the selection of an established category, then the category's prototype may be refined to incorporate new information in the input pattern. If the search ends by selecting a previously untrained node, the ART network establishes a new category.

Figure 1: ART 1

Figure 1 illustrates the main components of an ART 1 network and Figure 2 illustrates an ART search cycle. During ART search, an input vector **I** registers itself as a pattern **X** of activity across level F_1 (Figure 2a). Multiple converging and diverging $F_1 \rightarrow F_2$ adaptive filter pathways multiply the vector **S** by a matrix of adaptive weights, or long term memory (LTM) traces, to generate 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, strong competition selects the F_2 node that receives the maximal $F_1 \rightarrow F_2$ input. Only one component of **Y** is nonzero after this choice takes place. Activation of such a *winner-take-all* node defines the category, or symbol, of the input pattern **I**. Such a category represents all the inputs **I** that maximally activate the corresponding node.

Figure 2: ART Search

Activation of an F_2 node may be interpreted as "making a hypothesis" about an input **I**. An F_2 vector generates a signal vector **U** sent top-down through the $F_2 \rightarrow F_1$ adaptive filter. After multiplication by the adaptive weight matrix of the top-down filter, a vector **V** becomes the $F_2 \rightarrow F_1$ input (Figure 2b). Vector **V** plays the role of a learned top-down expectation. Activation of **V** by **Y** may be interpreted as "testing the hypothesis" **Y**, or "reading out the category prototype" **V**. The ART network matches the "expected prototype" **V** of the category against the active input pattern, or exemplar, **I**.

This matching process may change the F_1 activity pattern **X** by suppressing activation of all features in **I** that are not confirmed by **V**. The resultant pattern $\mathbf{X}^*$ encodes the pattern of features to which the network "pays attention". If the expectation **V** is close enough to the input **I**, then a state of *resonance* occurs, with the matched pattern **X** defining an attentional focus. The resonant state persists long enough for learning to occur; hence the term *adaptive resonance* theory. ART learns prototypes rather than exemplars because weights encode the attended feature vector $\mathbf{X}^*$, rather than the input **I** itself.

A dimensionless parameter called *vigilance* defines the criterion of an acceptable match. Vigilance weighs how close the input exemplar **I** must be to the top-down prototype **V** in order for resonance to occur. In ARTMAP, vigilance becomes an internally controlled variable, rather than a fixed parameter. Because vigilance can vary across learning trials, a single ART system can encode widely differing degrees of generalization, or morphological variability. Low vigilance leads to broad generalization, coarse categories, and abstract prototypes. High vigilance leads to narrow generalization, fine categories, and specific prototypes. In the limit of very high vigilance, prototype learning reduces to exemplar learning. Varying vigilance levels allow a single ART system to recognize both abstract categories of 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 **I** are too novel, or unexpected, to satisfy the vigilance criterion. Search leads to selection of a better recognition code, symbol, category, or hypothesis to represent input **I** at level F_2. An *orienting subsystem A* controls the search process. The orienting subsystem interacts with the attentional subsystem, as in Figures 2c and 2d, to enable the attentional subsystem to learn about novel inputs without risking unselective forgetting of its previous knowledge.

ART search prevents associations from forming between **Y** and $\mathbf{X}^*$ if $\mathbf{X}^*$ is too different from **I** 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 prototype may be refined in light of new information carried by **I**. If **I** is too different from any of the previously learned prototypes, then the search ends upon an uncommitted F_2 node, which begins a new category.

An ART *choice parameter* controls how deeply the search proceeds before selecting an uncommitted node. As learning self-stabilizes, all inputs coded by a category access it directly and search is automatically disengaged. The category selected is, then, the one whose prototype provides the globally best match to the input pattern. Stable on-line learning proceeds with familiar inputs directly activating their categories and novel inputs triggering adaptive searches, until the network's memory reaches its capacity. Simulations illustrate fuzzy ART dynamics in a parameter range called the *conservative limit*. In this limit, the choice parameter α (Figure 3) is very small. Then an input first selects a category whose weight vector is a fuzzy 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.

Figure 3: ART 1/fuzzy ART

Fuzzy ART

Fuzzy ART inherits the design features of other ART models. Figure 3 summarizes how the ART 1 operations of category choice, matching, search, and learning translate into fuzzy ART operations when the intersection operator ($\cap$) of ART 1 replaces the fuzzy intersection, or component-wise minimum, operator ($\wedge$). Despite this close formal homology, this chapter summarizes fuzzy ART as an algorithm, rather than as a locally defined neural model. Carpenter, Grossberg, and Rosen (1991b) describe a neural network realization of fuzzy ART. For the special case of binary inputs and fast learning, the computations of fuzzy ART are identical to those of the ART 1 neural network.

Fast-learn slow-recode and complement coding options: Many applications of ART 1 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 backpropagation. 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 backpropagation restricts its domain to off-line applications with a slow learning rate. In

addition, lacking the key feature of competition, a backpropagation 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 inputs that occur only rarely. Slow recoding then prevents features in a category's prototype from being erroneously deleted in response to noisy or partial inputs. Only a statistically persistent change in a feature's relevance to an established category can delete it from the prototype of the category.

Complement coding is a preprocessing step that normalizes input patterns. Complement coding solves a potential fuzzy ART category proliferation problem (Carpenter, Grossberg, and Rosen, 1991a; Moore, 1989). 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 prototype encodes features that are critically present in category exemplars, while the off-cell portion encodes features that are critically absent. Small weights in both on-cell and off-cell portions of a prototype encode as "uninformative" those features that are sometimes present and sometimes absent. In set theoretic terms, complement coding leads to a symmetric ART theory in which the fuzzy intersection ($\wedge$) and the fuzzy union ($\vee$) play complementary roles. Complement coding allows a geometric interpretation of fuzzy ART recognition categories as box-shaped regions of input space. Fuzzy intersections and unions iteratively define the corners of each box. Simulations in this chapter illustrate fuzzy ART geometry for an example where inputs are two-dimensional, so boxes are rectangles. The geometric formulation allows comparison between fuzzy ART and aspects of Salzberg's (1990) Nested Generalized Exemplars (NGE) and Simpson's (1991) Fuzzy Min-Max Classifier (FMMC).

Figure 4: ARTMAP

Fuzzy ARTMAP

Each ARTMAP system includes a pair of Adaptive Resonance Theory modules (ART_a and ART_b) that create stable recognition categories in response to arbitrary sequences of input patterns (Figure 4). During supervised learning, ART_a receives a stream $\{\mathbf{a}^{(p)}\}$ of input patterns and ART_b receives a stream $\{\mathbf{b}^{(p)}\}$ of input patterns, where $\mathbf{b}^{(p)}$ is the correct prediction given $\mathbf{a}^{(p)}$. 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 that 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.

An ART_a *baseline vigilance* parameter $\overline{\rho_a}$ calibrates the minimum confidence needed for ART_a to accept a chosen category, rather than search for a better one through automatically controlled search. Lower values of $\overline{\rho_a}$ enable larger categories to form, maximizing code compression. Initially, $\rho_a = \overline{\rho_a}$. During training, a predictive failure at ART_b increases ρ_a by the minimum amount needed to trigger ART_a search, through a feedback control mechanism called *match tracking* (Carpenter, Grossberg, and Reynolds, 1991). Match tracking sacrifices

the minimum amount of compression necessary to correct the predictive error. Hypothesis testing selects a new ART_a category, which focuses attention on a cluster of $\mathbf{a}^{(p)}$ input features that is better able to predict $\mathbf{b}^{(p)}$. 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.

An ARPA benchmark simulation, circle-in-the-square, illustrates fuzzy ARTMAP dynamics. The simulation task is learning to identify which points lie inside and which lie outside a circle. During training, components of the ART_a input $\mathbf{a}$ are the x- and y-coordinates of a point in the unit square; and ART_b input equals 0 or 1, identifying $\mathbf{a}$ as inside or outside the circle. As fuzzy ARTMAP learns on-line, or incrementally, test set accuracy increases from 88.6% to 98.0% as the training set increases in size from 100 to 100,000 randomly chosen points. With off-line learning, the system needs from 2 to 13 epochs to learn all training set exemplars to 100% accuracy, where an epoch is one cycle of training on an entire set of input exemplars. Test set accuracy then increases from 89.0% to 99.5% as the training set size increases from 100 to 100,000. Application of the voting strategy improves an average single-run accuracy of 90.5% on five runs to a voting accuracy of 93.9%, with each run trained on a fixed 1,000-item set for one epoch.

Fuzzy ART Algorithm

ART field activity vectors: Each ART system includes a field F_0 of nodes that represent a current input vector; a field F_1 that receives both bottom-up input from F_0 and top-down input from a field F_2 that represents the active code, or category (Figure 1). Vector $\mathbf{I} = (I_1, \ldots, I_M)$ denotes F_0 activity, with each component I_i in the interval [0,1], for $i = 1, \ldots, M$. Vector $\mathbf{x} = (x_1, \ldots, x_M)$ denotes F_1 activity and $\mathbf{y} = (y_1, \ldots, y_N)$ denotes F_2 activity. The number of nodes in each field is arbitrary.

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

$$w_{j1}(0) = \ldots = w_{jM}(0) = 1; \tag{1}$$

then each category is *uncommitted.* After a category codes its first input it becomes *committed.* Each component w_{ji} can decrease but never increase during learning. Thus each weight vector $\mathbf{w}_j(t)$ converges to a limit. The fuzzy ART weight, or prototype, vector $\mathbf{w}_j$ subsumes both the bottom-up and top-down weight vectors of ART 1 (Figure 1).

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.

Category choice: For each input $\mathbf{I}$ and F_2 node j, the *choice function* T_j is defined by

$$T_j(\mathbf{I}) = \frac{|\mathbf{I} \wedge \mathbf{w}_j|}{\alpha + |\mathbf{w}_j|}, \tag{2}$$

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

$$(\mathbf{p} \wedge \mathbf{q})_i \equiv \min(p_i, q_i) \tag{3}$$

and where the norm $|\cdot|$ is defined by

$$|\mathbf{p}| \equiv \sum_{i=1}^{M} |p_i|. \tag{4}$$

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

$$T_J = \max\{T_j : j = 1 \ldots N\}. \tag{5}$$

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 $j \neq J$. In a choice system, the F_1 activity vector $\mathbf{x}$ obeys the equation

$$\mathbf{x} = \begin{cases} \mathbf{I} & \text{if } F_2 \text{ is inactive} \\ \mathbf{I} \wedge \mathbf{w}_J & \text{if the } J^{th} \ F_2 \text{ node is chosen.} \end{cases} \tag{6}$$

Resonance or reset: *Resonance* occurs if the *match function* $|\mathbf{I} \wedge \mathbf{w}_J|/|\mathbf{I}|$ of the chosen category meets the vigilance criterion:

$$\frac{|\mathbf{I} \wedge \mathbf{w}_J|}{|\mathbf{I}|} \geq \rho; \tag{7}$$

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

$$|\mathbf{x}| = |\mathbf{I} \wedge \mathbf{w}_J| \geq \rho|\mathbf{I}|. \tag{8}$$

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

$$\frac{|\mathbf{I} \wedge \mathbf{w}_J|}{|\mathbf{I}|} < \rho; \tag{9}$$

that is, if

$$|\mathbf{x}| = |\mathbf{I} \wedge \mathbf{w}_J| < \rho|\mathbf{I}|. \tag{10}$$

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 (5). The search process continues until the chosen J satisfies the matching criterion (7).

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

$$\mathbf{w}_J^{(\text{new})} = \beta(\mathbf{I} \wedge \mathbf{w}_J^{(\text{old})}) + (1 - \beta)\mathbf{w}_J^{(\text{old})}. \tag{11}$$

Fast learning corresponds to setting $\beta = 1$. The learning law of the NGE system (Salzberg, 1990) is equivalent to (11) in the fast-learn limit with the complement coding.

Fast-commit, slow-recode: For efficient coding of noisy input sets, it is useful to set $\beta = 1$ when J is an uncommitted node, and then to take $\beta < 1$ for slower adaptation after the

category is already committed. The fast-commit, slow-recode option makes $\mathbf{w}_J^{(\text{new})} = \mathbf{I}$ the first time category J becomes active. Moore (1989) introduced the learning law (11), with fast commitment and slow recoding, to investigate a variety of generalized ART 1 models. Some of these models are similar to fuzzy ART, but none uses complement coding. Moore describes a category proliferation problem that can occur in some analog ART systems when many random inputs erode the norm of weight vectors. Complement coding solves this problem.

Normalization by complement coding: Normalization of fuzzy ART inputs prevents category proliferation. The $F_0 \rightarrow F_1$ inputs are normalized if, for some $\gamma > 0$,

$$\sum_i I_i = |\mathbf{I}| \equiv \gamma \tag{12}$$

for all inputs $\mathbf{I}$. One way to normalize each vector $\mathbf{a}$ is:

$$\mathbf{I} = \frac{\mathbf{a}}{|\mathbf{a}|}. \tag{13}$$

Complement coding normalizes the input but it also preserves amplitude information, in contrast to (13). Complement coding represents both the on-response and the off-response to an input vector $\mathbf{a}$ (Figure 4). In its simplest form, $\mathbf{a}$ represent the on-response and $\mathbf{a}^c$, the complement of $\mathbf{a}$, represents the off-response, where

$$a_i^c \equiv 1 - a_i. \tag{14}$$

The complement coded $F_0 \rightarrow F_1$ input $\mathbf{I}$ is the 2M-dimensional vector

$$\mathbf{I} = (\mathbf{a}, \mathbf{a}^c) \equiv (a_1, \ldots, a_M, a_1^c, \ldots, a_M^c). \tag{15}$$

A complement coded input is automatically normalized, because

$$\begin{aligned} |\mathbf{I}| &= |(\mathbf{a}, \mathbf{a}^c)| \\ &= \sum_{i=1}^{M} a_i + (M - \sum_{i=1}^{M} a_i) \\ &= M. \end{aligned} \tag{16}$$

With complement coding, the initial condition

$$w_{j1}(0) = \ldots = w_{j,2M}(0) = 1. \tag{17}$$

replaces the fuzzy ART initial condition (1).

The close linkage between fuzzy subsethood and ART choice/search/learning forms the foundation of the computational properties of fuzzy ART. In the conservative limit, where the choice parameter $\alpha = 0^+$, the choice function T_j measures the degree to which $\mathbf{w}_j$ is a fuzzy subset of $\mathbf{I}$ (Kosko, 1986). A category J for which $\mathbf{w}_J$ is a fuzzy subset of $\mathbf{I}$ will then be selected first, if such a category exists. Resonance depends on the degree to which $\mathbf{I}$ is a

fuzzy subset of $\mathbf{w}_J$, by (7) and (9). When J is such a fuzzy subset choice, then the match function value is:

$$\frac{|\mathbf{I} \wedge \mathbf{w}_J|}{|\mathbf{I}|} = \frac{|\mathbf{w}_J|}{|\mathbf{I}|}. \tag{18}$$

Choosing J to maximize $|\mathbf{w}_j|$ among fuzzy subset choices, by (2), thus maximizes the opportunity for resonance in (7). If reset occurs for the node that maximizes $|\mathbf{w}_J|$, then reset will also occur for all other subset choices.

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 $\mathbf{a}$. Consider an input set that consists of 2-dimensional vectors $\mathbf{a}$. With complement coding,

$$\mathbf{I} = (\mathbf{a}, \mathbf{a}^c) = (a_1, a_2, 1 - a_1, 1 - a_2). \tag{19}$$

Each category j then has a geometric representation as a rectangle R_j. Following (19), a complement-coded weight vector $\mathbf{w}_j$ takes the form:

$$\mathbf{w}_j = (\mathbf{u}_j, \mathbf{v}_j^c), \tag{20}$$

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 5a). The size of R_j is:

$$|R_j| \equiv |\mathbf{v}_j - \mathbf{u}_j|, \tag{21}$$

which is equal to the height plus the width of R_j Figure 5a.

Figure 5: Fuzzy ART boxes

In a fast-learn fuzzy ART system, with $\beta = 1$ in (11), $\mathbf{w}_J^{(\text{new})} = \mathbf{I} = (\mathbf{a}, \mathbf{a}^c)$ when J is an uncommitted node. The corners of $R_J^{(\text{new})}$ are then $\mathbf{a}$ and $(\mathbf{a}^c)^c = \mathbf{a}$. Hence $R_J^{(\text{new})}$ is just the point $\mathbf{a}$. Learning increases the size of R_J, which grows as the size of $\mathbf{w}_J$ shrinks during learning. Vigilance ρ determines the maximum size of R_J, with $|R_j| \leq 2(1 - \rho)$, as shown below. During each fast-learning trial, R_J expands to $R_J \oplus \mathbf{a}$, the minimum rectangle containing R_J and $\mathbf{a}$ (Figure 5b). 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 (3); and the fuzzy union $\vee$ is defined by:

$$(\mathbf{p} \vee \mathbf{q})_i \equiv \max(p_i, q_i) \tag{22}$$

(Zadeh, 1965). Hence, by (21), the size of $R_J \oplus \mathbf{a}$ is:

$$|R_J \oplus \mathbf{a}| = |(\mathbf{a} \vee \mathbf{v}_J) - (\mathbf{a} \wedge \mathbf{u}_J)|. \tag{23}$$

However, before R_J can expand to include $\mathbf{a}$, reset chooses another category if $|R_J \oplus \mathbf{a}|$ is too large. With fast learning, R_j is the smallest rectangle 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 vector is:

$$(\wedge_j \mathbf{a})_i = \min\{a_i : \mathbf{a} \text{ has been coded by category } j\} \tag{24}$$

and

$$(\vee_j \mathbf{a})_i = \max\{a_i : \mathbf{a} \text{ has been coded by category } j\} \tag{25}$$

(Figure 6). The size of R_j is

$$|R_j| = |\vee_j \mathbf{a} - \wedge_j \mathbf{a}| \tag{26}$$

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

$$\mathbf{w}_j = (\wedge_j \mathbf{a}, (\vee_j \mathbf{a})^c), \tag{27}$$

as in (20) and (21). Thus

$$|\mathbf{w}_j| = \sum_i (\wedge_j \mathbf{a})_i + \sum_i [1 - (\vee_j \mathbf{a})_i] = M - |\vee_j \mathbf{a} - \wedge_j \mathbf{a}|, \tag{28}$$

so the size of the box R_j is

$$|R_j| = M - |\mathbf{w}_j|. \tag{29}$$

By (8), (11), and (16),

$$|\mathbf{w}_j| \geq \rho M. \tag{30}$$

By (29) and (30),

$$|R_j| \leq (1 - \rho) M. \tag{31}$$

Figure 6: Fuzzy ART category

Inequality (31) shows that high vigilance ($\rho \cong 1$) leads to small R_j while low vigilance ($\rho \cong 0$) permits large R_j. If j is an uncommitted node, $|\mathbf{w}_j| = 2M$, by (17), so formally, $|R_j| \equiv -M$, by (29). These observations are combined into the following summary of fuzzy ART dynamics.

Fuzzy ART stable category learning: A fuzzy ART system with complement coding, fast learning, and constant vigilance forms categories that converge to limits in response to an arbitrary sequence of analog or binary input vectors. Category boxes can grow in each dimension, but never shrink. The size of a box R_j equals $M - |\mathbf{w}_j|$, where $\mathbf{w}_j$ is the corresponding weight vector. The size $|R_j|$ is bounded above by $M(1-\rho)$. In the conservative limit, one-pass learning obtains such that no reset or additional learning occurs on subsequent presentations of any input. Moreover, if $0 \leq \rho < 1$, the number of categories is bounded, even if the number of exemplars in the training set is unbounded. Similar properties hold for the fast-learn slow-recode case, except that repeated presentations of each input may be needed before stabilization occurs, even in the conservative limit.

A comparison of fuzzy ARTMAP, NGE, and FMMC: The geometry of the fuzzy boxes R_j resembles parts of the Nested Generalized Exemplar (NGE) system (Salzberg, 1990) and the Fuzzy Min-Max Classifier (FMMC) system (Simpson, 1991). Fuzzy ARTMAP, NGE, and FMMC all use boxes to represent category weights in a supervised learning paradigm. All three systems use some version of the learning law (11) to update weights when an input correctly predicts the output. The three algorithms differ, however, in their responses to incorrect predictions. Because NGE and FMMC do not have components analogous to the ART vigilance parameter, these algorithms do not have internal control of box size. NGE does include a type of search process, but its rules differ from those of fuzzy ARTMAP. For example, when NGE makes a predictive error, it searches at most two categories before

creating a new one. NGE allows boxes to shrink as well as to grow, so the fuzzy ART stability properties do not obtain. For the NGE system, only the Greedy version, a leader algorithm that codes the first exemplar of each category, is necessarily stable. Stability and match tracking allow fuzzy ARTMAP to construct automatically as many categories as are needed to learn any consistent training set to 100% accuracy. Both ARTMAP and NGE rely on multi-layer structures to effect their learning strategies. In contrast, the FMMC is a two-layer, feedforward system that allows at most one category box to represent each output class. FMMC can therefore learn only a limited set of category structures. In contrast, fuzzy ARTMAP can associate multiple categories with the same output, so, for example capital letters, script letters, and various fonts can all predict the same letter name.

Fuzzy ARTMAP Algorithm

Fuzzy ARTMAP incorporates two fuzzy ART modules ART_a and ART_b that are linked together via an inter-ART module F^{ab} called a *map field*. The map field forms predictive associations between categories and realizes the ARTMAP *match tracking rule*. Match tracking increases the ART_a vigilance parameter ρ_a in response to a predictive error, or mismatch, at ART_b. Match tracking reorganizes category structure so that subsequent presentations of the input do not repeat the error. An outline of the ARTMAP algorithm follows.

ART_a and ART_b: Inputs to ART_a and ART_b are complement coded. For ART_a, $\mathbf{I} = \mathbf{A} = (\mathbf{a}, \mathbf{a}^c)$; and for ART_b, $\mathbf{I} = \mathbf{B} = (\mathbf{b}, \mathbf{b}^c)$ (Figure 4). Variables in ART_a or ART_b are designated by subscripts or superscripts "a" or "b". For ART_a, $\mathbf{x}^a \equiv (x_1^a \ldots x_{2M_a}^a)$ denotes the F_1^a output vector; $\mathbf{y}^a \equiv (y_1^a \ldots y_{N_a}^a)$ denotes the F_2^a output vector; and $\mathbf{w}_j^a \equiv (w_{j1}^a, w_{j2}^a, \ldots, w_{j,2M_a})$ denotes the j^{th} ART_a weight vector. For ART_b, $\mathbf{x}^b \equiv (x_1^b \ldots x_{2M_b}^b)$ denotes the F_1^b output vector; $\mathbf{y}^b \equiv (y_1^b \ldots y_{N_b}^b)$ denotes the F_2^b output vector; and $\mathbf{w}_k^b \equiv (w_{k1}^b, w_{k2}^b, \ldots, w_{k,2M_b}^b)$ denotes the k^{th} ART_b weight vector. For the map field, $\mathbf{x}^{ab} \equiv (x_1^{ab}, \ldots, x_{N_b}^{ab})$ denotes the F^{ab} output vector, and $\mathbf{w}_j^{ab} \equiv (w_{j1}^{ab}, \ldots, w_{jN_b}^{ab})$ denotes the weight vector from the j^{th} F_2^a node to F^{ab}. Vectors $\mathbf{x}^a, \mathbf{y}^a, \mathbf{x}^b, \mathbf{y}^b$, and $\mathbf{x}^{ab}$ are reset to $\mathbf{0}$ between input presentations.

Map field activation: The map field F^{ab} receives input from either or both of the ART_a or ART_b category fields. A chosen F_2^a node J sends input to the map field F^{ab} via the weights $\mathbf{w}_J^{ab}$. An active F_2^b node K sends input to F^{ab} via one–to-one pathways between F_2^b and F^{ab}. If both ART_a and ART_b are active, then F^{ab} remains active only if ART_a predicts the same category as ART_b. The F^{ab} output vector $\mathbf{x}^{ab}$ obeys:

$$\mathbf{x}^{ab} = \begin{cases} \mathbf{y}^b \wedge \mathbf{w}_J^{ab} & \text{if the J}^{th}\ F_2^a \text{ node is active and } F_2^b \text{ is active} \\ \mathbf{w}_J^{ab} & \text{if the J}^{th}\ F_2^a \text{ node is active and } F_2^b \text{ is inactive} \\ \mathbf{y}^b & \text{if } F_2^a \text{ is inactive and } F_2^b \text{ is active} \\ \mathbf{0} & \text{if } F_2^a \text{ is inactive and } F_2^b \text{ is inactive.} \end{cases} \tag{32}$$

By (32), $\mathbf{x}^{ab} = \mathbf{0}$ if $\mathbf{y}^b$ fails to confirm the map field prediction made by $\mathbf{w}_J^{ab}$. Such a mismatch event triggers an ART_a search for a better category, as follows.

Match tracking: At the start of each input presentation ART_a vigilance ρ_a equals a baseline vigilance parameter $\overline{\rho_a}$. When a predictive error occurs, match tracking raises ART_a vigilance just enough to trigger a search for a new F_2^a coding node. ARTMAP detects a predictive error when

$$|\mathbf{x}^{ab}| < \rho_{ab}|\mathbf{y}^b|, \tag{33}$$

where ρ_{ab} is the map field vigilance parameter. A signal from the map field to the ART_a orienting subsystem causes ρ_a to "track the F_1^a match." That is, ρ_a increases until it is slightly higher than the F_1^a match value $|\mathbf{A} \wedge \mathbf{w}_J^a||\mathbf{A}|^{-1}$. Then, since

$$|\mathbf{x}^a| = |\mathbf{A} \wedge \mathbf{w}_J^a| < \rho_a|\mathbf{A}|, \tag{34}$$

ART_a fails to meet the matching criterion, as in (10), and the search for another F_2^a node begins. The search leads to an F_2^a node J with

$$|\mathbf{x}^a| = |\mathbf{A} \wedge \mathbf{w}_J^a| \geq \rho_a|\mathbf{A}| \tag{35}$$

and

$$|\mathbf{x}^{ab}| = |\mathbf{y}^b \wedge \mathbf{w}_J^{ab}| \geq \rho_{ab}|\mathbf{y}^b|. \tag{36}$$

If no such node exists and if all F_2^a nodes are already committed, F_2^a automatically shuts down for the remainder of the input presentation.

Map field learning: Weights w_{jk}^{ab} in $F_2^a \rightarrow F^{ab}$ paths initially satisfy

$$w_{jk}^{ab}(0) = 1. \tag{37}$$

During resonance with the ART_a category J active, $\mathbf{w}_J^{ab}$ approaches the map field vector $\mathbf{x}^{ab}$ as in (11). With fast learning, once J learns to predict the ART_b category K, that association is permanent; i.e., $w_{JK}^{ab} = 1$ for all time.

Fuzzy ARTMAP Simulation: Circle-in-the-Square

The circle-in-the square problem requires a system to predict which points of a square lie inside and which lie outside a circle whose area equals half that of the square (radius approximately 0.4). This task is a benchmark problem for performance evaluation in the ARPA Artificial Neural Network Technology (ANNT) Program (Wilensky, 1990). Wilensky examined 2-n-1 backpropagation systems on this problem. He studied systems where the number (n) of hidden units ranged from 5 to 100, and the corresponding number of weights ranged from 21 to 401. Training sets ranged in size from 150 to 14,000 points identified as in or out. To avoid over-fitting, training ended when accuracy on the training set reached 90%. Systems with 20 to 40 hidden units reached this criterion level most quickly, in about 5000 epochs. The trained systems then correctly classified approximately 90% of test set points.

Figure 7: Circle, 1-epoch
Figure 8: Circle, boxes

Figures 7 and 8 illustrate fuzzy ARTMAP performance on the circle-in-the-square task after one training epoch. As training set size increases from 100 exemplars (Figure 7a) to 100,000 exemplars (Figure 7d) the rate of correct test set predictions increases from 88.6% to 98.0% while the number of ART_a category nodes increases from 12 to 121. Each category node j requires four learned weights $\mathbf{w}_j^a$ in ART_a plus one map field weight $\mathbf{w}_j^{ab}$ to record whether category j predicts that a point lies inside or outside the circle. For example, 1-epoch training on 100 exemplars (Figure 7a) creates 12 ART_a categories and so uses 60 weights to achieve 88.6% test set accuracy. Figure 8 shows the ART_a category rectangles R_j^a for each

simulation of Figure 7. As in Figure 5 each rectangle R_j^a corresponds to the 4-dimensional weight vector $\mathbf{w}_j^a = (\mathbf{u}_j^a, (\mathbf{v}_j^a)^c)$, with $\mathbf{u}_j^a$ and $\mathbf{v}_j^a$ are plotted as the lower-left and upper-right corners of R_j^a, respectively. Early in training, large R_j^a estimate large areas as belonging to one or the other category (Figure 8a). Additional R_j^a improve accuracy, especially near the boundary of the circle (Figure 8d). The map becomes arbitrarily accurate provided the number of F_2^a nodes increases as needed.

Figure 9: Circle, multi-epochs

Figure 9 depicts the response patterns of fuzzy ARTMAP on another series of circle-in-the-square simulations that use the same training sets as in Figure 7. However, each input set is now presented for as many epochs as needed to achieve 100% predictive accuracy on the training set, whereas training in Figure 7 lasted for one epoch only. In each case, test set predictive accuracy increases, but so does the number of ART_a category nodes. For example, with 10,000 exemplars, 1-epoch training uses 50 ART_a nodes to give 96.7% test set accuracy (Figure 7c). The same training set, after 6 epochs, uses 39 more ART_a nodes to correct about half the errors and boost test set accuracy to 98.3% (Figure 9c).

Figure 7 shows how a test set error rate decreases from 11.4% to 2.0% as training set size increases from 100 to 100,000 in 1-epoch simulations. Figure 9 shows how test set error rates decrease further if learning continues for as many epochs as necessary to reach 100% accuracy on each training set. The ARTMAP voting strategy provides a third way to reduce test set errors. Recall that the voting strategy assumes a fixed set of training exemplars. Before each individual simulation, inputs are randomly ordered. After each simulation, the prediction of each test set item is recorded. The final prediction is the one made by the largest number of individual simulations. Voting almost always reduces errors, and the number of votes cast for a given outcome also attaches a measure of predictive confidence to each test set point.

Figure 10: Circle, voting

Simulations in Figure 10 show how, given a limited training set, voting across a few input orderings improves predictive accuracy by a factor that is comparable to the improvement attained by an order of magnitude increase in the size of a training set. In Figure 10, a fixed set of 1,000 randomly chosen exemplars is presented to a fuzzy ARTMAP system on five independent 1-epoch circle-in-the-square simulations. After each simulation, inside/outside predictions are recorded on a 1,000-item test set. Accuracy on individual simulations ranges from 85.9% to 93.4%, averaging 90.5%, and using from 15 to 23 ART_a nodes. Voting by the five simulations improves test set accuracy to 93.9% (Figure 10c). In other words, test set errors decrease from an average individual rate of 9.5% to a voting rate of 6.1%. Figure 10d, which indicates the number of votes cast for each test set point, reflects variations in predictive confidence across different regions, with confidence lowest near the border of the circle. Voting by more than five simulations maintains an error rate between 5.8% and 6.1%. This limit on further improvement by voting appears to be due to gaps in the fixed 1,000-item training set. By comparison, a tenfold increase in the size of the training set reduces the error by an amount similar to that achieved by five-simulation voting.

ART Applications

Since the publication of the first ART network in 1987, scientists and engineers have applied these systems to a variety of problems. Researchers often cite unique ART features

such as code stability, speed, and incremental learning as reasons for using ART or ARTMAP instead of an error-based neural network such as backpropagation.

The Boeing Company Neural Information Retrieval System (NIRS) has advanced from prototype to implementation in a state-of-the-art computer-aided airplane design system (Caudell, 1993; Smith, Escobedo, and Caudell, 1993). Engineers now use NIRS for production of the Boeing 747 and 767 airplanes and for design of the Boeing 777. The Neural Information Retrieval System is a hierarchy of ART networks that form compressed content-addressable memories of 2-D and 3-D parts designs. The NIRS shows an engineer who has sketched a part on the CAD system other parts in inventory that may be similar. Inventory proliferation and design time are both saved. Working CAD systems that include the NIRS have already reduced parts inventories by a factor of nine, and Boeing estimates that this technology will save the company up to $80 million per year. A new book, **Neural Networks in Design and Manufacturing** (Kumara *et al.*, 1993), discusses commercial applications of ART networks.

A trained ARTMAP system translates into a set of if-then rules at any stage of learning. This feature has made the network particularly useful in the analysis of large medical databases (Carpenter and Tan, 1993; Goodman *et al.*, 1992; Ham and Han, 1993; Harvey, 1993). Other ART medical applications include electrocardiogram wave recognition (Suzuki, Abe, and Ono, 1993). ARTMAP test set performance has proved superior to that of other neural networks in application domains such as diagnostic monitoring of nuclear plants (Keyvan, Dung, and Rabelo, 1993), land cover classification from remotely sensed data (Gopal, Sklarew, and Lambin, 1993), and the prediction of protein secondary structure (Mehta, Vij, and Rabelo, 1993). The ART-EMAP network adds to fuzzy ARTMAP spatial and temporal evidence accumulation capabilities (Carpenter and Ross, 1993). These new functions improve performance on both noisy and noise-free test sets, and expand the range of ARTMAP applications to spatio-temporal recognition problems such as 3-D object recognition and scene analysis. Researchers at MIT Lincoln Laboratory use ART systems for robot sensory-motor systems (Bachelder, Waxman, and Seibert, 1993; Baloch and Waxman, 1991), 3-D object recognition (Seibert and Waxman, 1991, 1992), and face recognition (Seibert and Waxman, 1993). The Macintosh commercial software Open Sesame! uses an unsupervised ART network to adapt the operating system to a user's work habits (Johnson, 1993). Other applications range from analyses of musical structure (Gjerdingen, 1990) to military target recognition (Moya, Koch, and Hostetler, 1993). Finally, applications of ART networks continue to include those of the original adaptive resonance theory: to organize, clarify, and predict neural and psychological data concerning memory recognition, and attention (Carpenter and Grossberg, 1991, 1993; Desimone, 1992; Gochin, 1990).

REFERENCES

Bachelder, I.A., Waxman, A.M., and Seibert, M. (1993). A neural system for mobile robot visual place learning and recognition. **Proceedings of the world congress on neural networks (WCNN-93)**, Hillsdale, NJ: Lawrence Erlbaum Associates, **I**-512–517.

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

Carpenter, G.A. and Gjaja, M. (1993). Generalized fuzzy ART. Technical Report CAS/CNS-TR-93-060. Boston, MA: Boston University.

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

Carpenter, G.A. and Grossberg, S. (1987b). ART 2: Stable self-organization of pattern recognition codes for analog input patterns. *Applied Optics*, **26**, 4919–4930.

Carpenter, G.A. and Grossberg, S. (1990). ART 3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures. *Neural Networks*, **3**, 129–152.

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

Carpenter, G.A. and Grossberg, S. (1993). Normal and amnesic learning, recognition, and memory by a neural model of cortio-hippocampal interactions. *Trends in Neurosciences*, **16**, 131–137.

Carpenter, G.A., Grossberg, S., Markuzon, N., Reynolds, J.H., and 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. Technical Report CAS/CNS-91-016. Boston, MA: Boston University.

Carpenter, G.A., Grossberg, S. and Reynolds, J.H. (1991). ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. *Neural Networks*, **4**, 565–588. Technical Report CAS/CNS-TR-91-001. Boston, MA: Boston University.

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

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

Carpenter, G.A., Grossberg, S., and Ross, W.D. (1993). ART-EMAP: A neural network architecture for learning and prediction by evidence accumulation. **Proceedings of the world congress on neural networks (WCNN-93)**, Hillsdale, NJ: Lawrence Erlbaum Associates, **III**-649–656. Technical Reports CAS/CNS-TR-93-015 and 93-035, Boston, MA: Boston University.

Carpenter, G.A., and Tan, A.-H. (1993). Rule extraction, fuzzy ARTMAP, and medical databases. **Proceedings of the world congress on neural networks (WCNN-93)**, Hillsdale, NJ: Lawrence Erlbaum Associates, **I**-501–506. Technical Report CAS/CNS-TR-93-016, Boston, MA: Boston University.

Caudell, T. (Ed.) (1993). **Adaptive neural systems.** 1992 IR&D Technical Report BCS-CS-ACS-93-008, Seattle, WA: The Boeing Company.

Desimone, R. (1992). Neural circuits for visual attention in the primate brain. In G.A. Carpenter and S. Grossberg (Eds.), **Neural networks for vision and image processing.** Cambridge, MA: MIT Press, pp. 343–364.

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

Gochin, P. (1990). Pattern recognition in primate temporal cortex: But is it ART? **Proceedings of the international joint conference on neural networks (IJCNN-90)** (Washington, DC), **I**-77–80. Hillsdale, NJ: Erlbaum Associates.

Goodman, P.H., Kaburlasos, V.G., Egbert, D.D., Carpenter, G.A., Grossberg, S., Reynolds, J.H., Rosen, D.B., and Hartz, A.J. (1992). Fuzzy ARTMAP neural network compared to linear discriminant analysis prediction of the length of hospital stay in patients with pneumonia. **Proceedings of the IEEE international conference on systems, man, and cybernetics** (Chicago, October, 1992). New York, NY: IEEE Press, **I**-748–753.

Gopal, S., Sklarew, D., and Lambin, E. (1993). Fuzzy-neural network classification of land cover change in the Sahel. **Proceedings of the DOSES/EUROSAT workshop on new tools for spatial analysis.** Lisbon, Portugal, November 18–20, 1993.

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

Ham, F.M. and Han, S.W. (1993). Quantitative study of the QRS complex using fuzzy ARTMAP and the MIT/BIH arrhythmia database. **Proceedings of the world congress on neural networks (WCNN-93)**, Hillsdale, NJ: Lawrence Erlbaum Associates, **I**-207–211.

Harvey, R.M. (1993). Nursing diagnosis by computers: An application of neural networks. *Nursing Diagnosis*, **4**, 26–34.

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

Keyvan, S., Durg, A., and Rabelo, L.C. (1993). Application of artificial neural networks for development of diagnostic monitoring system in nuclear plants. *American Nuclear Society Conference Proceedings*, April 18–21, 1993.

Kosko, B. (1986). Fuzzy entropy and conditioning. *Information Sciences*, **40**, 165–174.

Kumar, S.S. and Guez, A. (1991). ART based adaptive pole placement for neurocontrollers. *Neural Networks*, **4**, 319–335.

Kumara, S.R.T., Merchawi, N.S., Karmarthi, S.V., and Thazhutaveetil, M. (1993). **Neural networks in design and manufacturing.** Chapman and Hall Publishers.

Mehta, B.V., Vij, L., and Rabelo, L.C. (1993). Prediction of secondary structures of proteins using fuzzy ARTMAP. **Proceedings of the world congress on neural networks (WCNN-93)**, Hillsdale, NJ: Lawrence Erlbaum Associates, **I**-228–232.

Moore, B. (1989). ART 1 and pattern clustering. In D. Touretzky, G. Hinton, and T. Sejnowski (Eds.), **Proceedings of the 1988 connectionist models summer school.** San Mateo, CA: Morgan Kaufmann Publishers, pp. 174–185.

Moya, M.M., Koch, M.W., and Hostetler, L.D. (1993). One-class classifier networks for target recognition applications. **Proceedings of the world congress on neural networks (WCNN-93)**, Hillsdale, NJ: Lawrence Erlbaum Associates, **III**-797–801.

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

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

Rumelhart, D.E., Hinton, G. and Williams, R. (1986). Learning internal representations by error propagation. In D.E. Rumelhart and J.L. McClelland (Eds.). **Parallel distributed processing**. Cambridge, MA: MIT Press, pp. 318–362.

Salzberg, S.L. (1990). **Learning with nested generalized exemplars**. Boston, MA: Kluwer Academic Publishers.

Seibert, M. and Waxman, A.M. (1991). Learning and recognizing 3D objects from multiple views in a neural system. In H. Wechsler (Ed.), **Neural networks for perception, Volume 1**. New York: Academic Press.

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

Seibert, M. and Waxman, A.M. (1993). An approach to face recognition using saliency maps and caricatures. **Proceedings of the world congress on neural networks (WCNN-93)**, Hillsdale, NJ: Lawrence Erlbaum Associates, **III**-661–664.

Simpson, P. (1991). Fuzzy min-max classification with neural networks. *Heuristics: The Journal of Knowledge Engineering*, **4**, 1–9.

Smith, S.D.G., Escobedo, R., and Caudell, T.P. (1993). An industrial strength neural network application. **Proceedings of the world congress on neural networks (WCNN-93)**, Hillsdale, NJ: Lawrence Erlbaum Associates, **I**-490–494.

Suzuki, Y., Abe, Y., and Ono, K. (1993). Self-organizing QRS wave recognition system in ECG using ART 2. **Proceedings of the world congress on neural networks (WCNN'93)**, Hillsdale, NJ: Lawrence Erlbaum Associates, **IV**-39–42.

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

Wilensky, G. (1990). Analysis of neural network issues: Scaling, enhanced nodal processing, comparison with standard classification. DARPA Neural Network Program Review, October 29–30.

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

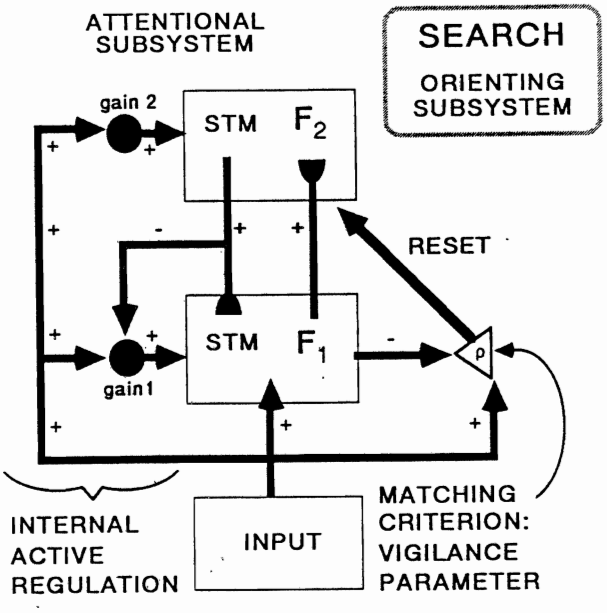


Figure 1. Typical ART 1 neural network (Carpenter and Grossberg, 1987a).

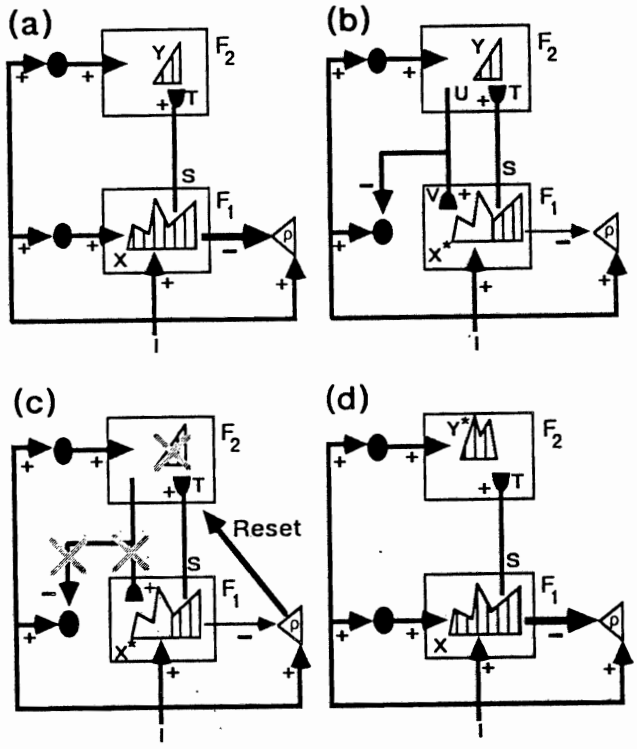


Figure 2. ART search for an F_2 code: (a) The input pattern **I** generates the specific STM activity pattern **X** at F_1 as it nonspecifically activates the orienting subsystem A. Pattern **X** both inhibits A and generates the output signal pattern **S**. An adaptive filter transforms the signal pattern **S** into the pattern **T**, which activates the STM pattern **Y** across F_2. (b) Pattern **Y** generates the signal pattern **U**, and a top-down adaptive filter transforms **U** into the prototype pattern **V**. If **V** mismatches **I**, then F_1 registers a new STM activity pattern $\mathbf{X}^*$. The resulting reduction of total STM reduces the total inhibition from F_1 to A. (c) If the ART matching criterion fails, A releases a nonspecific signal that resets the STM pattern **Y** at F_2. (d) Since reset inhibits **Y**, it also eliminates the top-down prototype signal **T**, so **X** can be reinstated at F_1. Enduring traces of the prior reset allow **X** to activate a different STM pattern $\mathbf{Y}^*$ at F_2. If the top-down prototype due to $\mathbf{Y}^*$ also mismatches **I** at F_1, then the search for an F_2 code that satisfies the matching criterion continues.

ART 1 (BINARY)	FUZZY ART (ANALOG)
CATEGORY CHOICE	
$T_j = \frac{\lvert I \cap w_j \rvert}{\alpha + \lvert w_j \rvert}$	$T_j = \frac{\lvert I \wedge w_j \rvert}{\alpha + \lvert w_j \rvert}$
MATCH CRITERION	
$\frac{\lvert I \cap w_j \rvert}{\lvert I \rvert} \geq \rho$	$\frac{\lvert I \wedge w_j \rvert}{\lvert I \rvert} \geq \rho$
FAST LEARNING	
$w_j^{(new)} = I \cap w_j^{(old)}$	$w_j^{(new)} = I \wedge w_j^{(old)}$
$\cap$ = logical AND intersection	$\wedge$ = fuzzy AND minimum

Figure 3. Analogy between ART 1 and fuzzy ART. In ART 1 $\mathbf{w}_j$ denotes the index set of top-down LTM traces that exceed a prescribed positive threshold value.

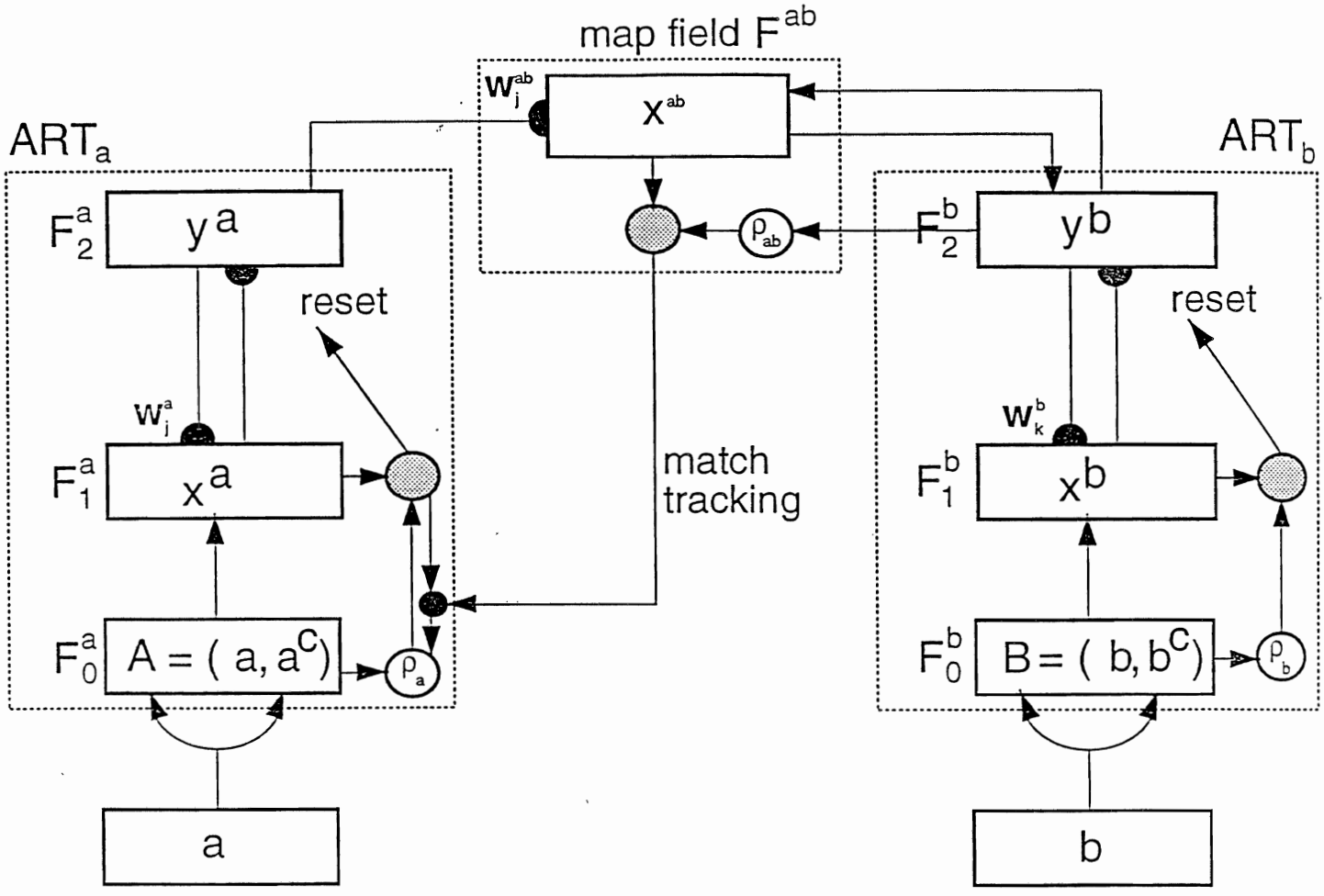


Figure 4. Fuzzy ARTMAP architecture. The ART_a complement coding preprocessor transforms the M_a-vector $\mathbf{a}$ into the $2M_a$-vector $\mathbf{A} = (\mathbf{a}, \mathbf{a}^c)$ at the ART_a field F_0^a. $\mathbf{A}$ is the input vector to the ART_a field F_1^a. Similarly, the input to F_1^b is the $2M_b$-vector $(\mathbf{b}, \mathbf{b}^c)$. 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 to activation of either an ART_a category that correctly predicts $\mathbf{b}$ or to a previously uncommitted ART_a category node.

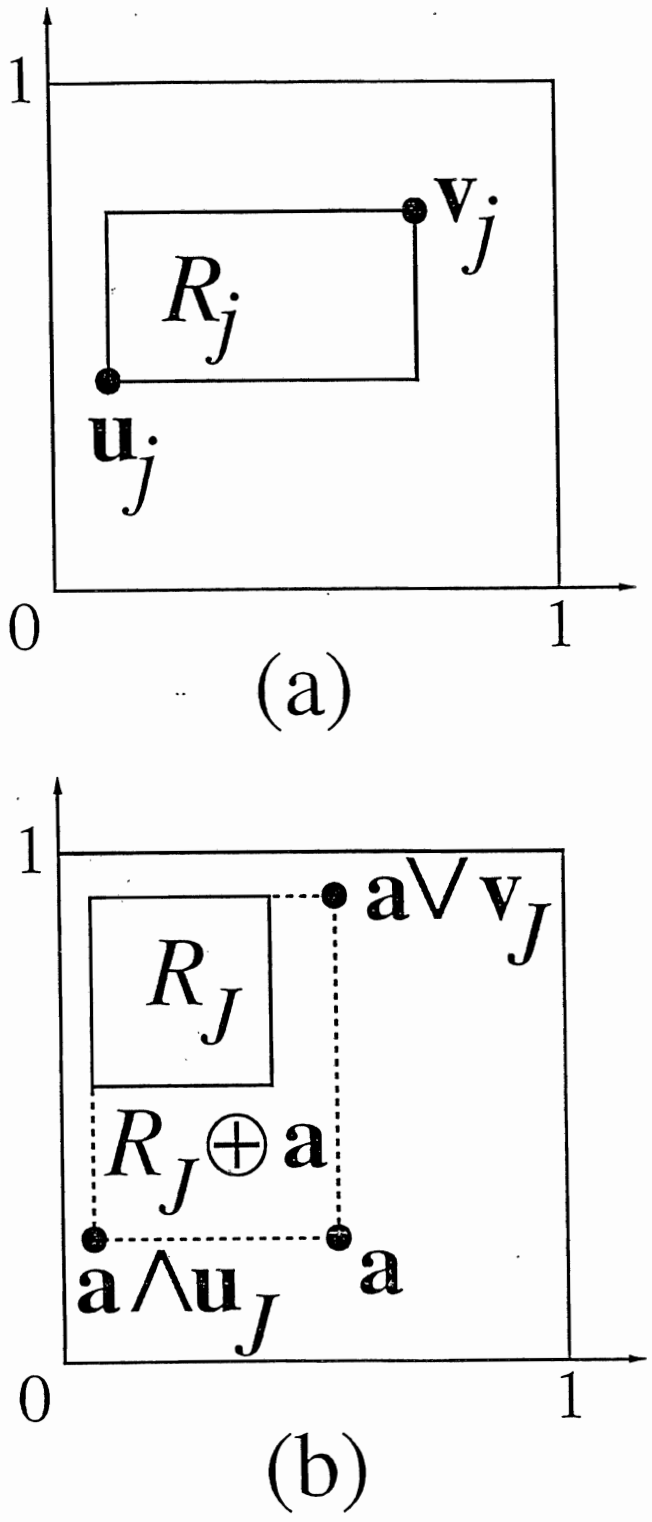


Figure 5. Fuzzy ART category boxes. (a) In complement coding form with $M = 2$, each weight vector $\mathbf{w}_j$ has a geometric interpretation as a rectangle R_j with corners $(\mathbf{u}_j, \mathbf{v}_j)$. (b) During fast learning, R_J expands to $R_J \oplus \mathbf{a}$, the smallest rectangle that includes R_J and $\mathbf{a}$, provided that $|R_J \oplus \mathbf{a}| \leq 2(1 - \rho)$.

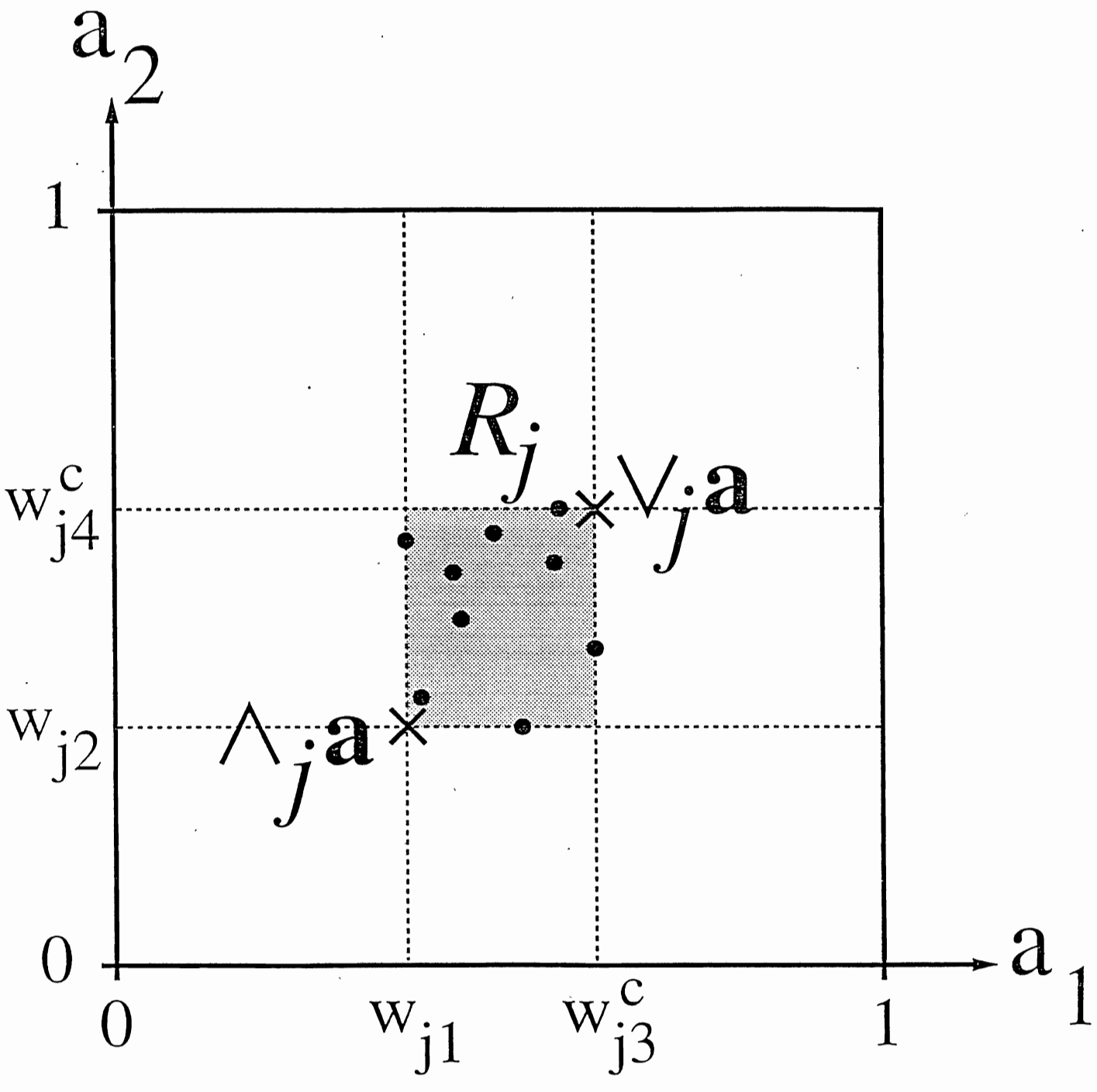
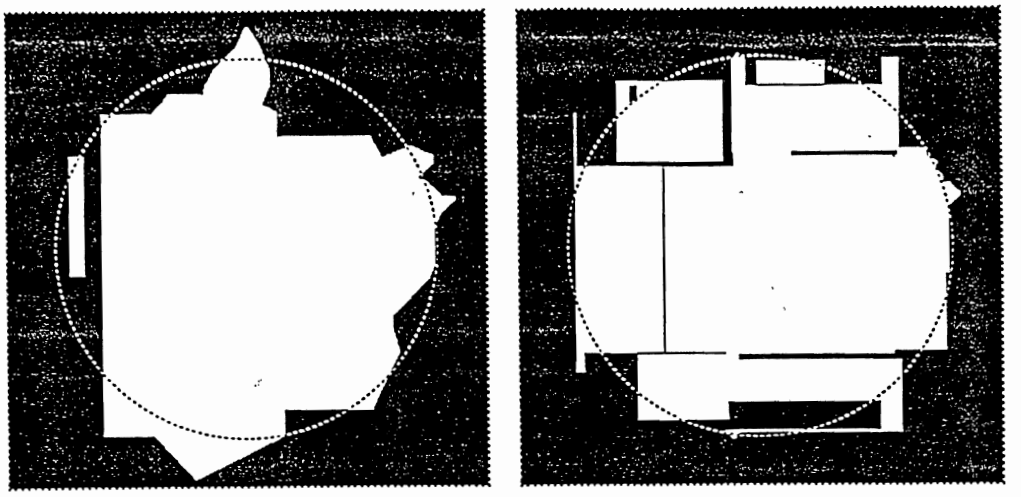
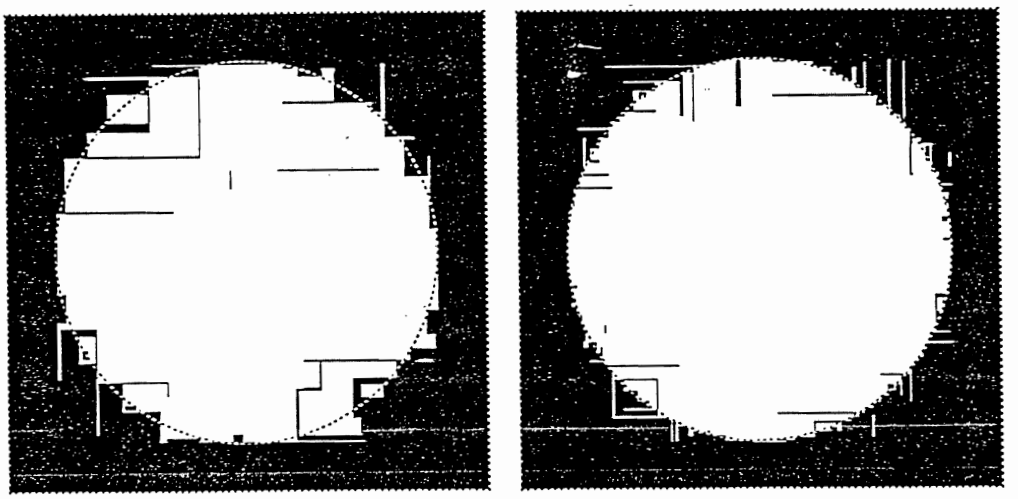


Figure 6. With fuzzy ART fast learning and complement coding, the j^{th} category rectangle R_j includes all those vectors $\mathbf{a}$ in the unit square that have activated category j without reset. The weight vector $\mathbf{w}_j$ equals $(\wedge_j \mathbf{a}, (\vee_j \mathbf{a})^c)$.



(a)
100 exemplars
99.0% training set
88.6% test set
12 ART_a categories

(b)
1,000 exemplars
95.5% training set
92.5% test set
21 ART_a categories



(c)
10,000 exemplars
97.7% training set
96.7% test set
50 ART_a categories

(d)
100,000 exemplars
98.8% training set
98.0% test set
121 ART_a categories

Figure 7. Circle-in-the-square test set response patterns after 1 epoch of fuzzy ARTMAP training on (a) 100, (b) 1,000, (c) 10,000, and (d) 100,000 randomly chosen training set points. The system predicts that test set points in white areas lie inside the circle and that points in black areas lie outside the circle. The test set error rate decreases, approximately inversely with the number of ART_a categories, as the training set size increases.

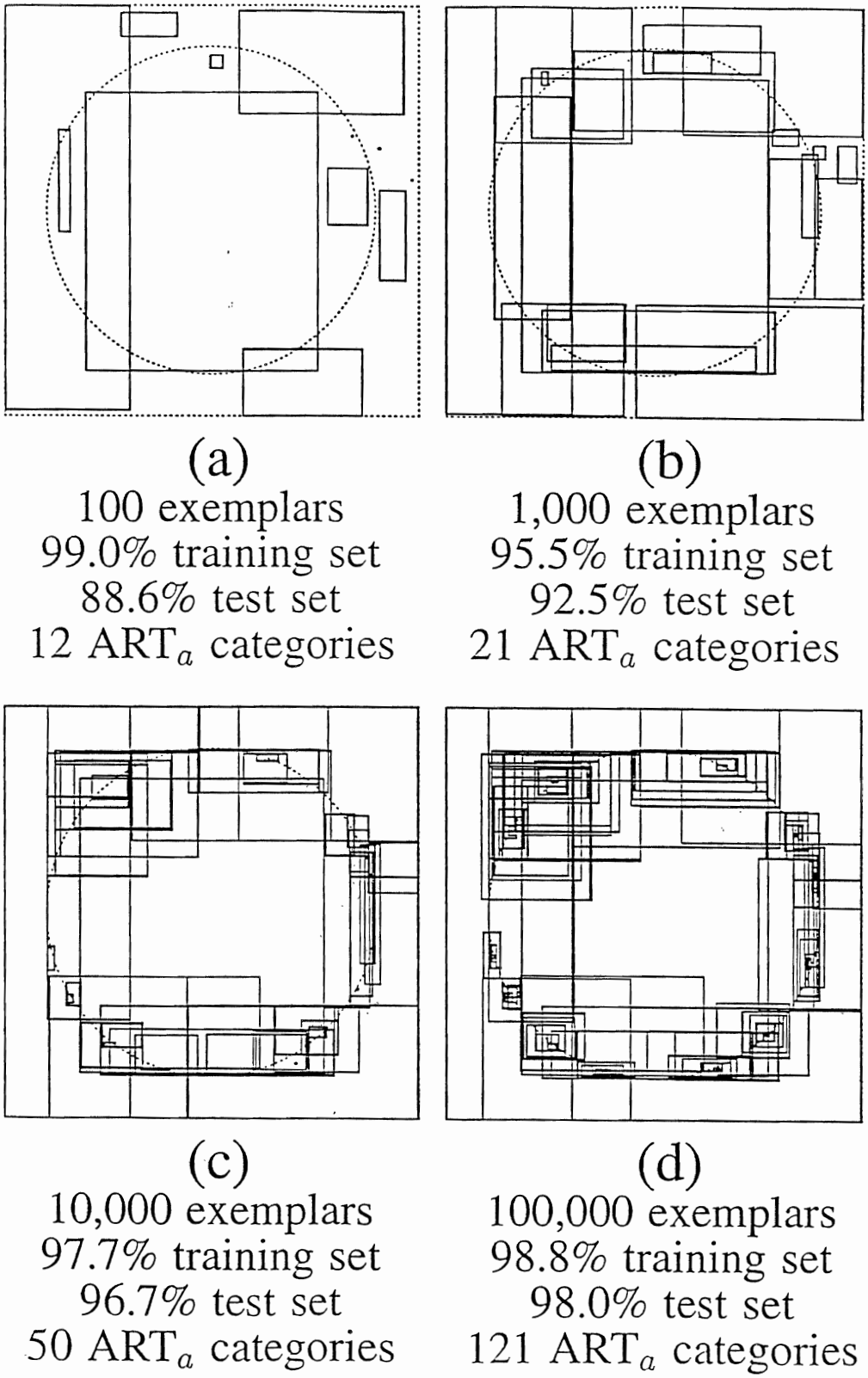
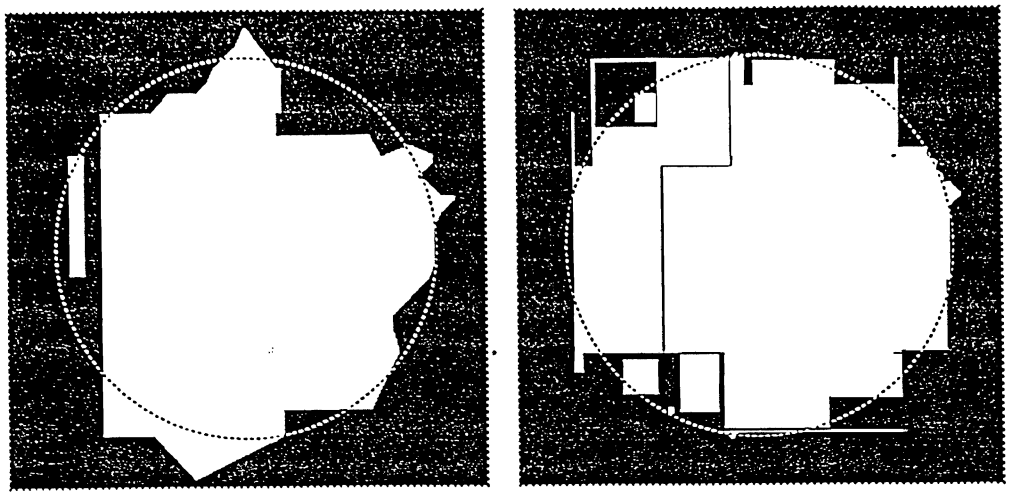
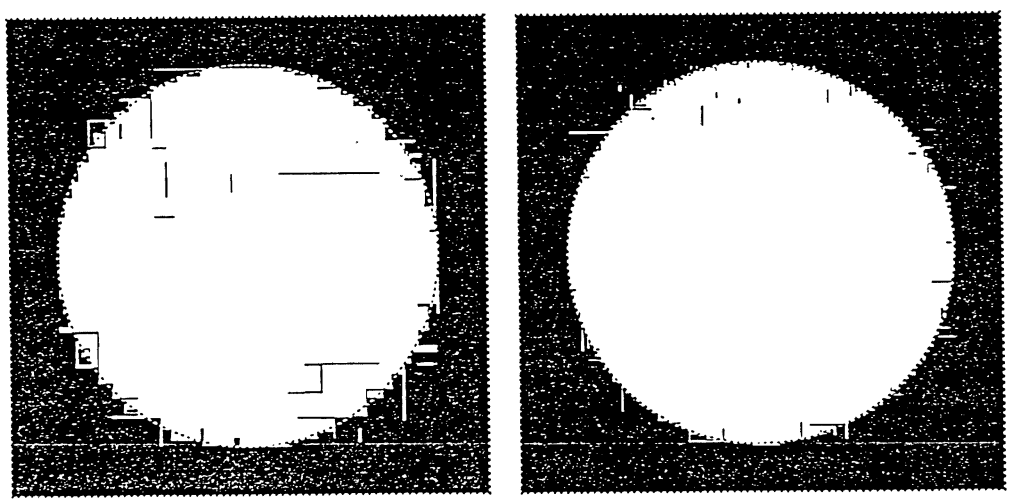


Figure 8. Fuzzy ARTMAP category rectangles R_j^a for the circle-in-the-square simulations of Figure 7. Small rectangles form near map discontinuities as the error rate drops toward 0.



(a)
100 exemplars
2 epochs
89.0% test set
12 ART_a categories

(b)
1,000 exemplars
3 epochs
95.0% test set
27 ART_a categories



(c)
10,000 exemplars
6 epochs
98.3% test set
89 ART_a categories

(d)
100,000 exemplars
13 epochs
99.5% test set
254 ART_a categories

Figure 9. Circle-in-the-square test set response patterns with exemplars repeatedly presented until the system achieves 100% correct prediction on (a) 100, (b) 1,000, (c) 10,000, and (d) 100,000 training set points. Training sets are the same as those used for Figures 7 and 8. Training to 100% accuracy requires (a) 2 epochs, (b) 3 epochs, (c) 6 epochs, and (d) 13 epochs. Additional training epochs decrease test set error rates but create additional ART_a categories, compared to the 1-epoch simulation in Figure 7.

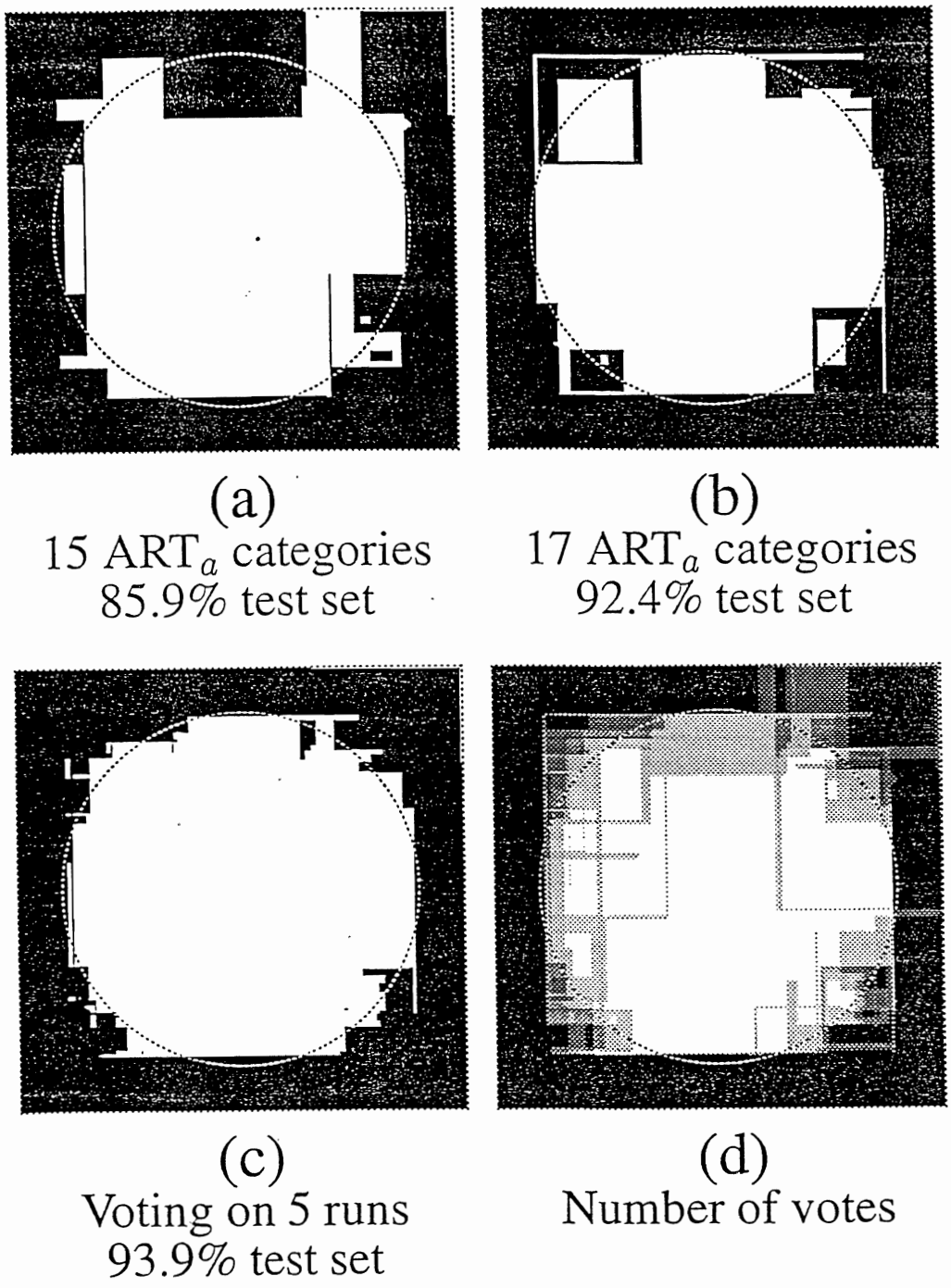


Figure 10. Circle-in-the-square response patterns for a fixed 1,000-item training set. (a) Test set responses after training on inputs presented in random order. After 1 epoch that uses 15 ART$_a$ nodes, test set prediction rate is 85.9%, the worst of 5 runs. (b) Test set responses after training on inputs presented in a different random order. After 1 epoch that uses 17 ART$_a$ nodes, test set prediction rate is 92.3%, the best of 5 runs. (c) Voting strategy applied to five individual simulations. Test set prediction rate is 93.9%. (d) Cumulative test set response pattern from five 1-epoch simulations. Gray scale intensity increases with the number of votes that predict a point to lie outside the circle.