ATTENTIVE SUPERVISED LEARNING AND RECOGNITION BY AN ADAPTIVE RESONANCE SYSTEM

Gail A. Carpenter†, Stephen Grossberg‡, Natalya Markuzon§,
John H. Reynolds¶, and David B. Rosen¶

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

1. Introduction

ARTMAP is a class of neural network architectures that employ attentional mechanisms to perform incremental supervised learning of recognition categories and multidimensional maps. The first ARTMAP system (Carpenter, Grossberg, and Reynolds, 1991) was used to classify binary vectors. This article describes a more general ARTMAP system that learns to classify analog as well as binary vectors (Carpenter, Grossberg, Markuzon, Reynolds, and Rosen, 1992). This generalization is accomplished by replacing the ART 1 modules (Carpenter and Grossberg, 1987a) of the binary ARTMAP system with Fuzzy ART modules (Carpenter, Grossberg, and Rosen, 1991a). Where ART 1 dynamics are described in terms of set-theoretic operations, Fuzzy ART dynamics are described in terms of fuzzy set-theoretic operations (Zadeh, 1965). Hence the new system is called Fuzzy ARTMAP. Also described is an ARTMAP *voting strategy*. This voting strategy is based on the observation that ARTMAP fast learning typically leads to different adaptive weights and recognition categories for different orderings of a given training set, even when overall predictive accuracy of all simulations is similar. The different category structures cause the set of test set items where errors occur to vary from one simulation to the next. The voting strategy uses an ARTMAP system that is trained several times on input sets 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 cancels many of the errors. Further, the voting strategy can be used to assign confidence estimates to competing predictions given small, noisy, or incomplete training sets.

† Supported in part by British Petroleum (89-A-1204), DARPA (AFOSR 90-0083), the National Science Foundation (NSF IRI 90-00530) and the Office of Naval Research (ONR N00014-91-J-4100).

‡ Supported in part by the Air Force Office of Scientific Research (AFOSR 90-0175), DARPA (AFOSR 90-0083) and the Office of Naval Research (ONR N00014-91-J-4100).

§ Supported in part by National Science Foundation (NSF IRI 90-00530) and British Petroleum (89-A-1204).

¶ Supported in part by DARPA (AFOSR 90-0083).

Acknowledgements: The authors wish to thank Cynthia E. Bradford for her valuable assistance in the preparation of the manuscript.

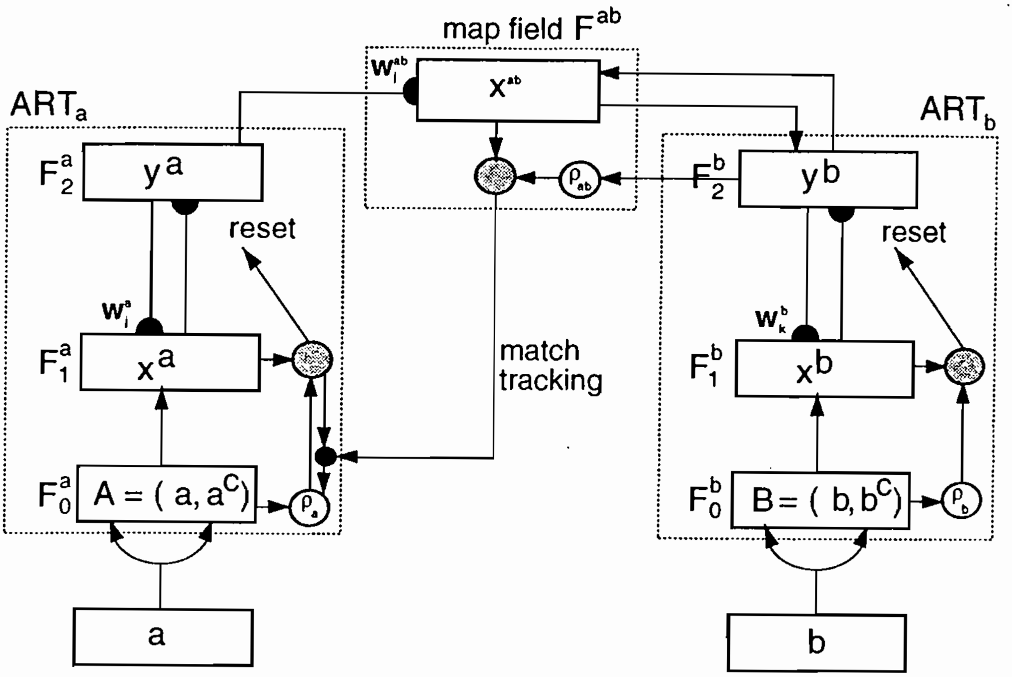


Figure 1. Fuzzy ARTMAP architecture. The ART_a complement coding preprocessor transforms the M_a-vector **a** into the $2M_a$-vector $\mathbf{A} = (\mathbf{a}, \mathbf{a}^c)$ at the ART_a field F_0^a. **A** is the input vector to the ART_a field F_1^a. Similarly, the input to F_1^b is the $2M_b$-vector $(\mathbf{b}, \mathbf{b}^c)$. When a prediction by ART_a is disconfirmed at ART_b, inhibition of map field activation 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 **b** or to a previously uncommitted ART_a category node.

Simulations illustrate Fuzzy ARTMAP performance as compared to benchmark back propagation and genetic algorithm systems. In all cases, Fuzzy ARTMAP simulations lead to favorable levels of learned predictive accuracy, speed, and code compression in both on-line and off-line settings. Fuzzy ARTMAP is also easy to use. It has a small number of parameters, requires no problem-specific system crafting or choice of initial weight values, and does not get trapped in local minima.

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 1). 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 $\mathbf{a}^{(p)}$. These modules are linked by an associative learning network and an internal controller that ensures autonomous system operation in real time. The controller is designed to create the minimal number of ART_a recognition categories, or "hidden units," needed to meet accuracy criteria. It does this by realizing a Minimax Learning Rule that enables an ARTMAP system to learn quickly, efficiently, and accurately as it conjointly *minimizes* predictive error and *maximizes* predictive generalization. This scheme automatically links predictive success to category size on a trial-by-trial basis using only local operations. It works by increasing the vigilance parameter ρ_a of ART$_a$ by the minimal amount needed to correct a predictive error at ART$_b$.

Parameter ρ_a calibrates the minimum confidence that ART$_a$ must have in a recognition category, or hypothesis, activated by an input $\mathbf{a}^{(p)}$ in order for ART$_a$ to accept that category, rather than search for a better one through an automatically controlled process of hypothesis testing. Lower values of ρ_a enable larger categories to form. These lower ρ_a values lead to broader generalization and higher code compression. A predictive failure at ART$_b$ increases ρ_a by the minimum amount needed to trigger hypothesis testing at ART$_a$, using a mechanism called *match tracking* (Carpenter, Grossberg, and Reynolds, 1991). Match tracking sacrifices the minimum amount of generalization necessary to correct a predictive error. Hypothesis testing leads to the selection of a new ART$_a$ category, which focuses attention on a new cluster of $\mathbf{a}^{(p)}$ input features that is better able to predict $\mathbf{b}^{(p)}$. Due to the combination of match tracking and fast learning, a single ARTMAP system can learn a different prediction for a rare event than for a cloud of similar frequent events in which it is embedded.

Whereas binary ARTMAP employs ART 1 systems for the ART$_a$ and ART$_b$ modules, Fuzzy ARTMAP substitutes Fuzzy ART systems for these modules. Fuzzy ART shows how computations from fuzzy set theory can be incorporated naturally into ART systems. For example, the intersection ($\cap$) operator that describes ART 1 dynamics is replaced by the AND operator ($\wedge$) of fuzzy set theory (Zadeh, 1965) in the choice, search, and learning laws of ART 1 (Figure 2). Especially noteworthy is the close relationship between the computation that defines fuzzy subsethood (Kosko, 1986) and the computation that defines category choice in ART 1. Replacing operation $\cap$ by operation $\wedge$ leads to a more powerful version of ART 1. Whereas ART 1 can learn stable categories only in response to binary input vectors, Fuzzy ART can learn stable categories in response to either analog or binary input vectors. Moreover, Fuzzy ART reduces to ART 1 in response to binary input vectors.

In Fuzzy ART, learning always converges because all adaptive weights are monotone nonincreasing. Without additional processing, this useful stability property could lead to the unattractive property of category proliferation as too many adaptive weights converge to zero. A preprocessing step, called complement coding, uses on-cell and off-cell responses to prevent category proliferation. Complement coding normalizes input vectors while preserving the amplitudes of individual feature activations. Without complement coding, an ART category memory encodes the degree to which critical features are consistently present in the training exemplars of that category. With complement coding, both the degree of absence and the degree of presence of features are represented by the category weight vector. The corresponding computations employ fuzzy OR ($\vee$, maximum) operators, as well as fuzzy AND ($\wedge$, minimum) operators.

This article includes summaries of the ART, Fuzzy ART, and Fuzzy ARTMAP systems. Section 2 describes the main characteristics of ART models, and Section 3 describes Fuzzy ART. Section 4 shows how two Fuzzy ART unsupervised learning modules are linked to form the Fuzzy ARTMAP supervised learning system. Sections 5 and 6 present two classes of benchmark simulation results. Section 5 describes a simulation task of learning to identify which points lie inside and which lie outside a given circle. Fuzzy ARTMAP on-line learning

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

Figure 2. Comparison of ART 1 and Fuzzy ART.

(also called incremental learning) is demonstrated, with test set accuracy increasing from 88.6% to 98.0% as the training set increased in size from 100 to 100,000 randomly chosen points. With off-line learning, the system needed from 2 to 13 epochs to learn all training set exemplars to 100% accuracy, where an epoch is defined as one cycle of training on an entire set of input exemplars. Test set accuracy then increased from 89.0% to 99.5% as the training set size increased from 100 to 100,000 points. Application of the voting strategy improved an average single-run accuracy of 90.5% on five runs to a voting accuracy of 93.9%, where each run trained on a fixed 1,000-item set for one epoch. These simulations are compared with studies by Wilensky (1990) of back propagation systems. These systems used at least 5,000 epochs to reach 90% accuracy on training and testing sets.

Section 6 describes Fuzzy ARTMAP performance on a benchmark letter recognition task developed by Frey and Slate (1991). Each database training exemplar represents a capital letter, in one of a variety of fonts and with significant random distortions, as a

16-dimensional feature vector. Each feature is assigned a value from 0 to 15. A number from 0 to 25 identifies the letters A–Z. Frey and Slate used this database to train a variety of classifiers that incorporate Holland-style genetic algorithms (Holland, 1980). Trained on 16,000 exemplars and tested on 4,000 exemplars, the best performing classifier had a test-set error rate of about 17.3%, more than three times the minimal error rate of an individual Fuzzy ARTMAP system (5.3%) and more than four times the error rate of a Fuzzy ARTMAP voting system (4.0%). In fact, application of the voting strategy improved an average accuracy of 93.9% on five separate runs to a voting accuracy of 96.0%. Moreover, this improved ARTMAP performance did not require greater memory resources: Fuzzy ARTMAP created fewer than 1,070 ART_a recognition categories in all simulations, compared to 1,040–1,302 rules created by the most accurate genetic algorithms.

2. ART Systems and Fuzzy Logic

Adaptive Resonance Theory, or ART, was introduced as a theory of human cognitive information processing (Grossberg, 1976, 1980). The theory has since led to an evolving series of real-time neural network models for unsupervised category learning and pattern recognition. These models are capable of learning stable recognition categories in response to arbitrary input sequences with either fast or slow learning. Model families include ART 1 (Carpenter and Grossberg, 1987a), which can stably learn to categorize binary input patterns presented in an arbitrary order; ART 2 (Carpenter and Grossberg, 1987b), which can stably learn to categorize either analog or binary input patterns presented in an arbitrary order; and ART 3 (Carpenter and Grossberg, 1990), which can carry out parallel search, or hypothesis testing, of distributed recognition codes in a multi-level network hierarchy. Variations of these models adapted to the demands of individual applications have been developed by a number of authors.

Figure 3 illustrates one example from the family of ART 1 models, and Figure 4 illustrates a typical ART search cycle. As shown in Figure 4a, an input vector **I** registers itself as a pattern **X** of activity across level F_1. The F_1 output vector **S** is then transmitted through the multiple converging and diverging adaptive filter pathways emanating from F_1. This transmission event multiplies 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**. A compressed activity vector **Y** is thereby generated across F_2. In ART 1, the competition is tuned so that the F_2 node that receives the maximal $F_1 \rightarrow F_2$ input is selected. 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.

Activation of an F_2 node may be interpreted as "making a hypothesis" about an input **I**. When **Y** is activated, it generates a signal vector **U** that is sent top-down through the second adaptive filter. After multiplication by the adaptive weight matrix of the top-down filter, a net vector **V** inputs to F_1 (Figure 4b). 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 1 network is designed to match 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

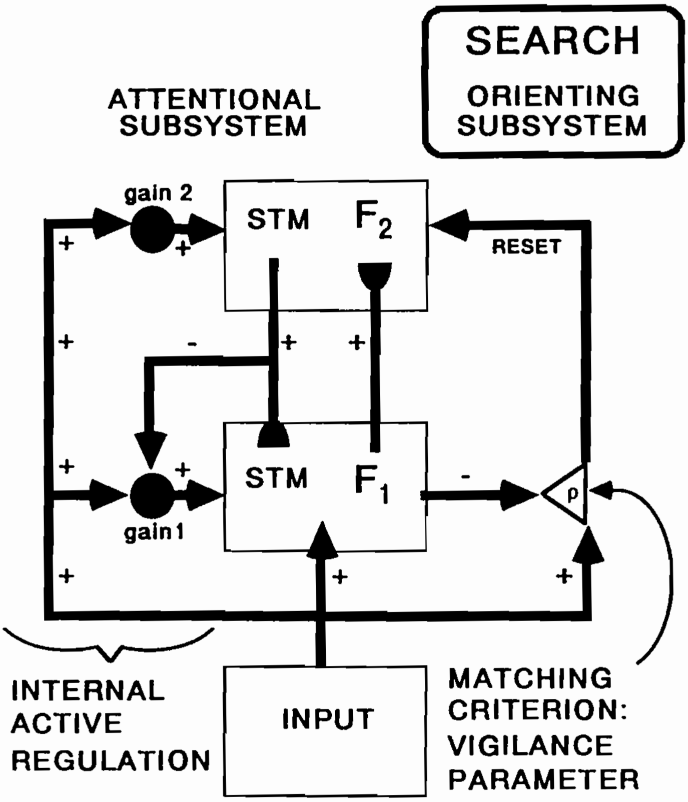


Figure 3. Typical ART 1 neural network (Carpenter and Grossberg, 1987a).

of all the feature detectors 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 as the attentional focus takes hold. The resonant state persists long enough for learning to occur; hence the term *adaptive resonance* theory. ART 1 learns prototypes, rather than exemplars, because the attended feature vector $\mathbf{X}^*$, rather than the input **I** itself, is learned.

The criterion of an acceptable match is defined by a dimensionless parameter called *vigilance*. Vigilance weighs how close the input exemplar **I** must be to the top-down prototype **V** in order for resonance to occur. Because vigilance can vary across learning trials, recognition categories capable of encoding widely differing degrees of generalization, or morphological variability, can be learned by a single ART system. Low vigilance leads to broad generalization and abstract prototypes. High vigilance leads to narrow generalization and to prototypes that represent fewer input exemplars. In the limit of very high vigilance, prototype learning reduces to exemplar learning. Thus a single ART system may be used, say, to recognize abstract categories of faces and dogs, as well as individual faces and dogs.

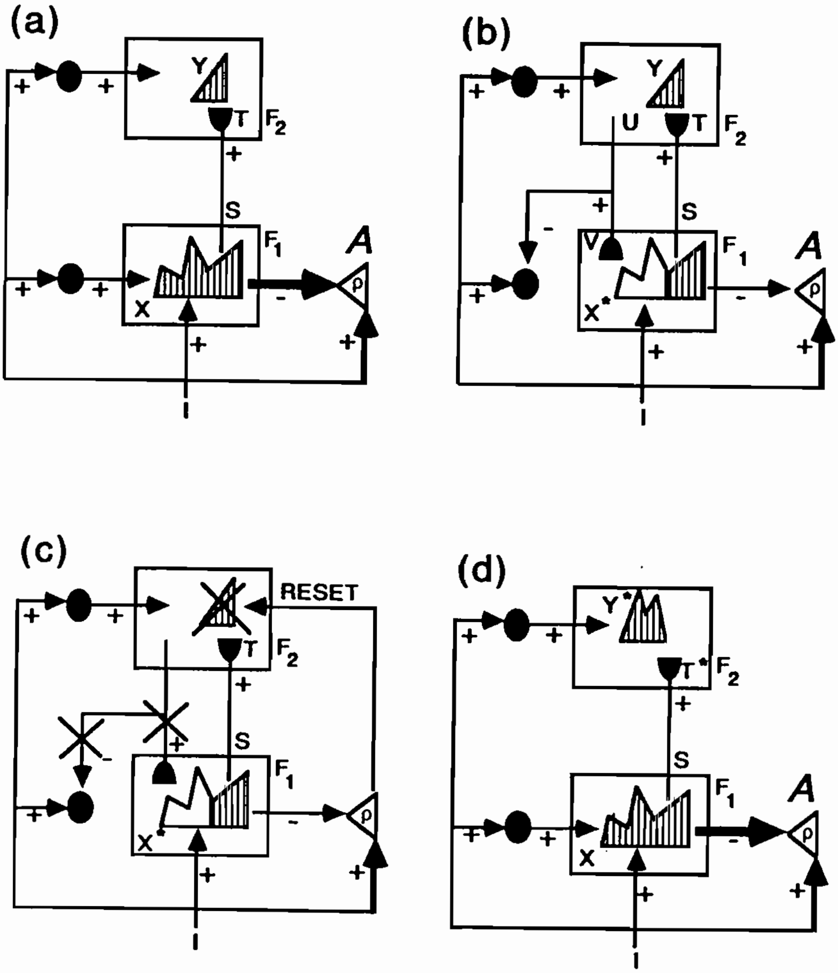


Figure 4. ART search for an F_2 code: (a) The input pattern **I** generates the specific STM activity pattern **X** at F_1 as it nonspecifically activates the orienting subsystem A. Pattern **X** both inhibits A and generates the output signal pattern **S**. Signal pattern **S** is transformed into the input pattern **T**, which activates the STM pattern **Y** across F_2. (b) Pattern **Y** generates the top-down signal pattern **U** which is transformed into the prototype pattern **V**. If **V** mismatches **I** at F_1, then a new STM activity pattern $\mathbf{X}^*$ is generated at F_1. The reduction in total STM activity which occurs when **X** is transformed into $\mathbf{X}^*$ causes a decrease in the total inhibition from F_1 to A. (c) If the matching criterion fails to be met, A releases a nonspecific arousal wave to F_2, which resets the STM pattern **Y** at F_2. (d) After **Y** is inhibited, its top-down prototype signal is eliminated, and **X** can be reinstated at F_1. Enduring traces of the prior reset lead **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 appropriate F_2 code continues.

If the top-down expectation **V** and the bottom-up input **I** are too novel, or unexpected, to satisfy the vigilance criterion, then a bout of hypothesis testing, or memory search, is triggered. Search leads to selection of a better recognition code, symbol, category, or hypothesis to represent input **I** at level F_2. An *orienting subsystem* mediates the search process (Figure 3). The orienting subsystem interacts with the attentional subsystem, as in Figures 4c and 4d, to enable the attentional subsystem to learn about novel inputs without risking unselective forgetting of its previous knowledge.

The search process 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. A familiar category may be selected by the search if its prototype is similar enough to the input **I** to satisfy the vigilance criterion. The prototype may then be refined in light of new information carried by **I**. If **I** is too different from any of the previously learned prototypes, then an uncommitted F_2 node is selected and learning of a new category is initiated.

A network parameter controls how deeply the search proceeds before an uncommitted node is chosen. As learning of a particular category self-stabilizes, all inputs coded by that category access it directly in a one-pass fashion, and search is automatically disengaged. The category selected is, then, the one whose prototype provides the globally best match to the input pattern. Learning can proceed on-line, and in a stable fashion, with familiar inputs directly activating their categories, while novel inputs continue to trigger adaptive searches for better categories, until the network's memory capacity is fully utilized.

The read-out of the top-down expectation **V** may be interpreted as a type of hypothesis-driven query. The matching process at F_1 and the hypothesis testing process at F_2 may be interpreted as query-driven symbolic substitutions. From this perspective, ART systems provide examples of new types of self-organizing production systems (Laird, Newell, and Rosenbloom, 1987). By incorporating predictive feedback into their control of the hypothesis testing cycle, ARTMAP systems embody self-organizing production systems that are also goal-oriented. ARTMAP systems are thus a new type of self-organizing expert system which is capable of stable autonomous fast learning about nonstationary environments that may contain a great deal of morphological variability. The fact that fuzzy logic may also be usefully incorporated into ARTMAP systems blurs even further the traditional boundaries between artificial intelligence and neural networks.

The Fuzzy ART model incorporates the design features of other ART models due to the close formal homolog between ART 1 and Fuzzy ART operations. Figure 2 summarizes how the ART 1 operations of category choice, matching, search, and learning translate into Fuzzy ART operations by replacing the set theory intersection operator ($\cap$) of ART 1 by the fuzzy set theory conjunction, or MIN operator ($\wedge$). Despite this close formal homology, Fuzzy ART is described as an algorithm, rather than a locally defined neural model. A neural network realization of Fuzzy ART is described elsewhere (Carpenter, Grossberg, and Rosen, 1991b). 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. The Fuzzy ART algorithm also includes two optional features, one concerning learning and the other input preprocessing, as described in Section 3.

3. Summary of the 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 3). The F_0 activity vector is denoted $\mathbf{I} = (I_1, \ldots, I_M)$, with each component I_i in the interval [0,1], $i = 1, \ldots, M$. The F_1 activity vector is denoted $\mathbf{x} = (x_1, \ldots, x_M)$ and the F_2 activity vector is denoted $\mathbf{y} = (y_1, \ldots, y_N)$. 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 LTM traces. Initially

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

then each category is said to be *uncommitted*. After a category is selected for coding it becomes *committed*. As shown below, each LTM trace w_{ji} is monotone nonincreasing through time and hence converges to a limit. The Fuzzy ART weight vector $\mathbf{w}_j$ subsumes both the bottom-up and top-down weight vectors of ART 1.

Parameters: Fuzzy ART dynamics are determined by a choice parameter $\alpha > 0$; a learning rate parameter $\beta \in [0,1]$; and a vigilance parameter $\rho \in [0,1]$.

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 AND (Zadeh, 1965) operator $\wedge$ 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

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

for any M-dimensional vectors $\mathbf{p}$ and $\mathbf{q}$. For notational simplicity, $T_j(\mathbf{I})$ in (2) is often written as T_j when the input $\mathbf{I}$ is fixed.

The system is said to make a *category choice* when at most one F_2 node can become active at a given time. The category choice is indexed by J, where

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

If more than one T_j is maximal, the category j with the smallest 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 is chosen, 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 is then chosen, by (5). The search process continues until the chosen J satisfies (7).

Learning: Once search ends, the weight vector $\mathbf{w}_J$ is updated 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 used in the EACH system of Salzberg (1990) is equivalent to equation (11) in the fast-learn limit with the complement coding option described below.

Fast-commit slow-recode option: For efficient coding of noisy input sets, it is useful to set $\beta = 1$ when J is an uncommitted node, and then to take $\beta < 1$ after the category is committed. Then $\mathbf{w}_J^{(\text{new})} = \mathbf{I}$ 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 includes the complement coding option. Moore described a category proliferation problem that can occur in some analog ART systems when a large number of inputs erode the norm of weight vectors. Complement coding solves this problem.

Input normalization/complement coding option: Proliferation of categories is avoided in Fuzzy ART if inputs are normalized. *Complement coding* is a normalization rule that preserves amplitude information. Complement coding represents both the on-response and the off-response to an input vector $\mathbf{a}$ (Figure 1). To define this operation in its simplest form, let $\mathbf{a}$ itself represent the on-response. The complement of $\mathbf{a}$, denoted by $\mathbf{a}^c$, represents the off-response, where

$$a_i^c \equiv 1 - a_i. \tag{12}$$

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

$$\mathbf{I} = (\mathbf{a}, \mathbf{a}^c) \equiv (a_1, \ldots, a_M, a_1^c, \ldots, a_M^c). \tag{13}$$

Note that

$$\begin{aligned}|\mathbf{I}| &= |(\mathbf{a},\mathbf{a}^c)| \\ &= \sum_{i=1}^{M} a_i + (M - \sum_{i=1}^{M} a_i) \\ &= M,\end{aligned} \tag{14}$$

so inputs preprocessed into complement coding form are automatically normalized. Where complement coding is used, the initial condition (1) is replaced by

$$w_{j1}(0) = \ldots = w_{j,2M}(0) = 1. \tag{15}$$

4. Fuzzy ARTMAP Algorithm

The Fuzzy ARTMAP system 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 is used to form predictive associations between categories and to realize the *match tracking rule* whereby the vigilance parameter of ART$_a$ increases in response to a predictive mismatch at ART$_b$. Match tracking reorganizes category structure so the predictive error is not repeated on subsequent presentations of the input. A circuit realization of the match tracking rule that uses only local real-time operations is provided in Carpenter, Grossberg, and Reynolds, (1991). The interactions mediated by the map field F^{ab} may be operationally characterized as follows.

ART$_a$ and ART$_b$

Inputs to ART$_a$ and ART$_b$ are in the complement code form: for ART$_a$, $\mathbf{I} = \mathbf{A} = (\mathbf{a},\mathbf{a}^c)$; for ART$_b$, $\mathbf{I} = \mathbf{B} = (\mathbf{b},\mathbf{b}^c)$ (Figure 1). Variables in ART$_a$ or ART$_b$ are designated by subscripts or superscripts "a" or "b". For ART$_a$, let $\mathbf{x}^a \equiv (x_1^a \ldots x_{2M_a}^a)$ denote the F_1^a output vector; let $\mathbf{y}^a \equiv (y_1^a \ldots y_{N_a}^a)$ denote the F_2^a output vector; and let $\mathbf{w}_j^a \equiv (w_{j1}^a, w_{j2}^a, \ldots, w_{j,2M_a})$ denote the j^{th} ART$_a$ weight vector. For ART$_b$, let $\mathbf{x}^b \equiv (x_1^b \ldots x_{2M_b}^b)$ denote the F_1^b output vector; let $\mathbf{y}^b \equiv (y_1^b \ldots y_{N_b}^b)$ denote the F_2^b output vector; and let $\mathbf{w}_k^b \equiv (w_{k1}^b, w_{k2}^b, \ldots, w_{k,2M_b}^b)$ denote the k^{th} ART$_b$ weight vector. For the map field, let $\mathbf{x}^{ab} \equiv (x_1^{ab}, \ldots, x_{N_b}^{ab})$ denote the F^{ab} output vector, and let $\mathbf{w}_j^{ab} \equiv (w_{j1}^{ab}, \ldots, w_{jN_b}^{ab})$ denote 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 set to $\mathbf{0}$ between input presentations.

Map field activation

The map field F^{ab} is activated whenever one of the ART$_a$ or ART$_b$ categories is active. If node J of F_2^a is chosen, then its weights $\mathbf{w}_J^{ab}$ activate F^{ab}. If node K in F_2^b is active, then the node K in F^{ab} is activated by 1–to–1 pathways between F_2^b and F^{ab}. If both ART$_a$ and ART$_b$ are active, then F^{ab} becomes active only if ART$_a$ predicts the same category as ART$_b$ via the weights $\mathbf{w}_J^{ab}$. The F^{ab} output vector $\mathbf{x}^{ab}$ obeys

$$\mathbf{x}^{ab} = \begin{cases} \mathbf{y}^b \wedge \mathbf{w}_J^{ab} & \text{if the Jth } F_2^a \text{ node is active and } F_2^b \text{ is active} \\ \mathbf{w}_J^{ab} & \text{if the Jth } F_2^a \text{ node is active and } F_2^b \text{ is inactive} \\ \mathbf{y}^b & \text{if } F_2^a \text{ is inactive and } F_2^b \text{ is active} \\ \mathbf{0} & \text{if } F_2^a \text{ is inactive and } F_2^b \text{ is inactive.} \end{cases} \tag{16}$$

By (16), $\mathbf{x}^{ab} = \mathbf{0}$ if the prediction $\mathbf{w}_J^{ab}$ is disconfirmed by $\mathbf{y}^b$. Such a mismatch event triggers an ART$_a$ search for a better category, as follows.

Match tracking

At the start of each input presentation the ART$_a$ vigilance parameter ρ_a equals a baseline vigilance $\overline{\rho_a}$. The map field vigilance parameter is ρ_{ab}. If

$$|\mathbf{x}^{ab}| < \rho_{ab}|\mathbf{y}^b|, \tag{17}$$

then ρ_a is increased until it is slightly larger than $|\mathbf{A} \wedge \mathbf{w}_J^a||\mathbf{A}|^{-1}$, where $\mathbf{A}$ is the input to F_1^a, in complement coding form. Then

$$|\mathbf{x}^a| = |\mathbf{A} \wedge \mathbf{w}_J^a| < \rho_a|\mathbf{A}|, \tag{18}$$

where J is the index of the active F_2^a node, as in (10). When this occurs, ART$_a$ search leads either to activation of another F_2^a node J with

$$|\mathbf{x}^a| = |\mathbf{A} \wedge \mathbf{w}_J^a| \geq \rho_a|\mathbf{A}| \tag{19}$$

and

$$|\mathbf{x}^{ab}| = |\mathbf{y}^b \wedge \mathbf{w}_J^{ab}| \geq \rho_{ab}|\mathbf{y}^b|; \tag{20}$$

or, if no such node exists, to the shut-down of F_2^a for the remainder of the input presentation.

Map field learning

Learning rules determine how the map field weights w_{jk}^{ab} change through time, as follows. Weights w_{jk}^{ab} in $F_2^a \to F^{ab}$ paths initially satisfy

$$w_{jk}^{ab}(0) = 1. \tag{21}$$

During resonance with the ART$_a$ category J active, $\mathbf{w}_J^{ab}$ approaches the map field vector $\mathbf{x}^{ab}$. 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.

5. Simulation: Circle-in-the-Square

The circle-in-the square problem requires a system to identify which points of a square lie inside and which lie outside a circle whose area equals half that of the square. This task was specified as a benchmark problem for system performance evaluation in the DARPA Artificial Neural Network Technology (ANNT) Program (Wilensky, 1990). Wilensky examined the performance of 2–n–1 back propagation 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. To avoid over-fitting, training was stopped when accuracy on the training set reached 90%. This criterion level was reached most quickly (5,000 epochs) in systems with 20 to 40 hidden units. In this condition, approximately 90% of test set points, as well as training set points, were correctly classified.

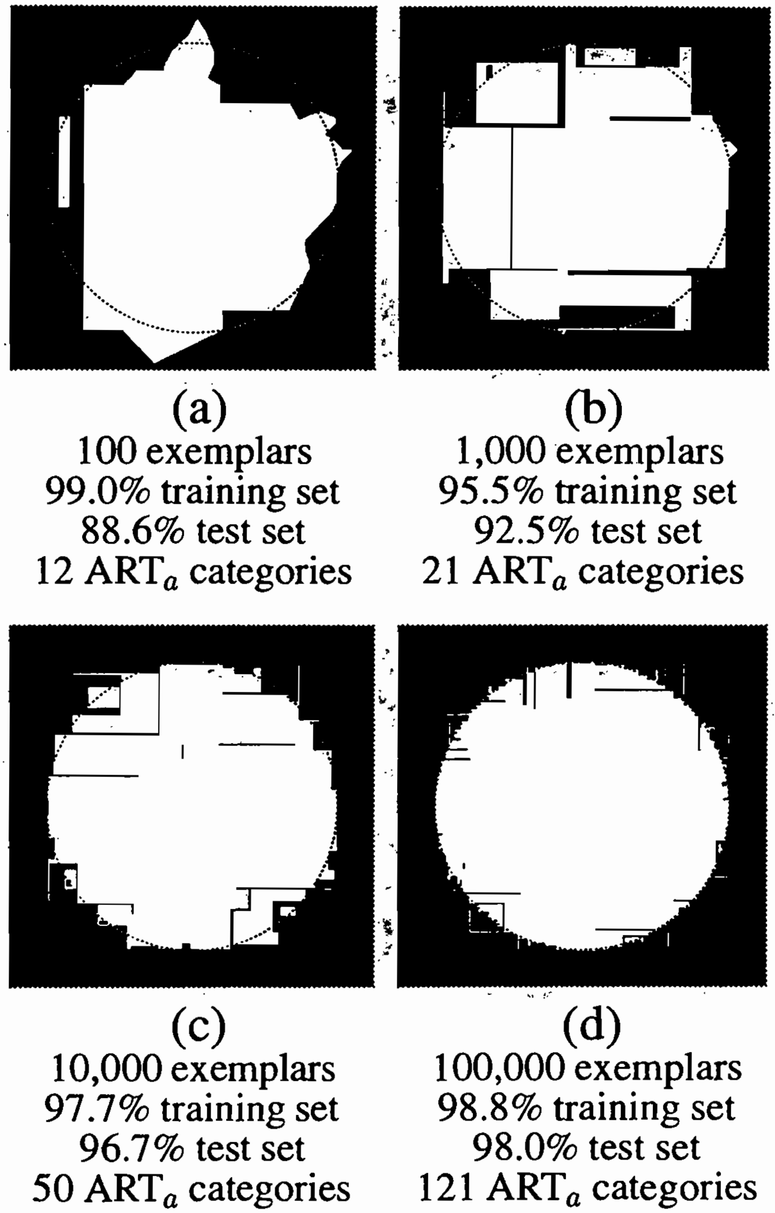
Fuzzy ARTMAP performance on this task after one training epoch is illustrated in Figures 5 and 6. As training set size increased from 100 exemplars (Figure 5a) to 100,000 exemplars (Figure 5d) the rate of correct test set predictions increased from 88.6% to 98.0% while the number of ART_a category nodes increased from 12 to 121. Each category node j required four learned weights $\mathbf{w}_j^a$ in ART_a plus one map field weight $\mathbf{w}_j$ to record whether category j predicts that a point lies inside or outside the circle. Thus, for example, 1-epoch training on 100 exemplars used 60 weights to achieve 88.6% test set accuracy. The map can be made arbitrarily accurate provided the number of ART_a nodes is allowed to increase as needed.

Figure 5 shows how a test set error rate is reduced from 11.4% to 2.0% as training set size increases from 100 to 100,000 in 1-epoch simulations. Test set error rate can be further reduced if exemplars are presented for as many epochs as necessary to reach 100% accuracy on the training set. The ARTMAP voting strategy provides a third way to eliminate test set errors. Recall that the voting strategy assumes a fixed set of training exemplars. Before each individual simulation the input ordering is randomly assembled. After each simulation the prediction of each test set item is recorded. Voting selects the outcome predicted by the largest number of individual simulations. In case of a tie, one outcome is selected at random. The number of votes cast for a given outcome provides a measure of predictive confidence at each test set point. Given a limited training set, voting across a few simulations can improve predictive accuracy by a factor that is comparable to the improvement that could be attained by an order of magnitude more training set inputs, as shown in the following example.

A fixed set of 1,000 randomly chosen exemplars was presented to a Fuzzy ARTMAP system on five independent 1-epoch circle-in-the-square simulations. After each simulation, inside/outside predictions were recorded on a 1,000-item test set. Accuracy on individual simulations ranged from 85.9% to 92.3%, averaging 90.5%; and the system used from 15 to 23 ART_a nodes. Voting by the five simulations improved test set accuracy to 93.9% (Figure 6c). In other words, test set errors were reduced from an average individual rate of 9.5% to a voting rate of 6.1%. Figure 6d indicates the number of votes cast for each test set point, and hence reflects variations in predictive confidence across different regions. Voting by more than five simulations maintained an error rate between 5.8% and 6.1%. This limit on further improvement by voting appears to be due to random gaps in the fixed 1,000-item training set. By comparison, a ten-fold increase in the size of the training set reduced the error by an amount similar to that achieved by five-simulation voting. For example, in Figure 5b, 1-epoch training on 1,000 items yielded a test set error rate of 7.5%; while increasing the size of the training set to 10,000 reduced the test set error rate to 3.3% (Figure 5c).

6. Simulation: Letter Image Recognition

Frey and Slate (1991) recently developed a benchmark machine learning task that they describe as a "difficult categorization problem" (p. 161). The task requires a system to identify an input exemplar as one of 26 capital letters A–Z. The database was derived from 20,000 unique black-and-white pixel images. The difficulty of the task is due to the wide variety of letter types represented: the twenty "fonts represent five different stroke styles



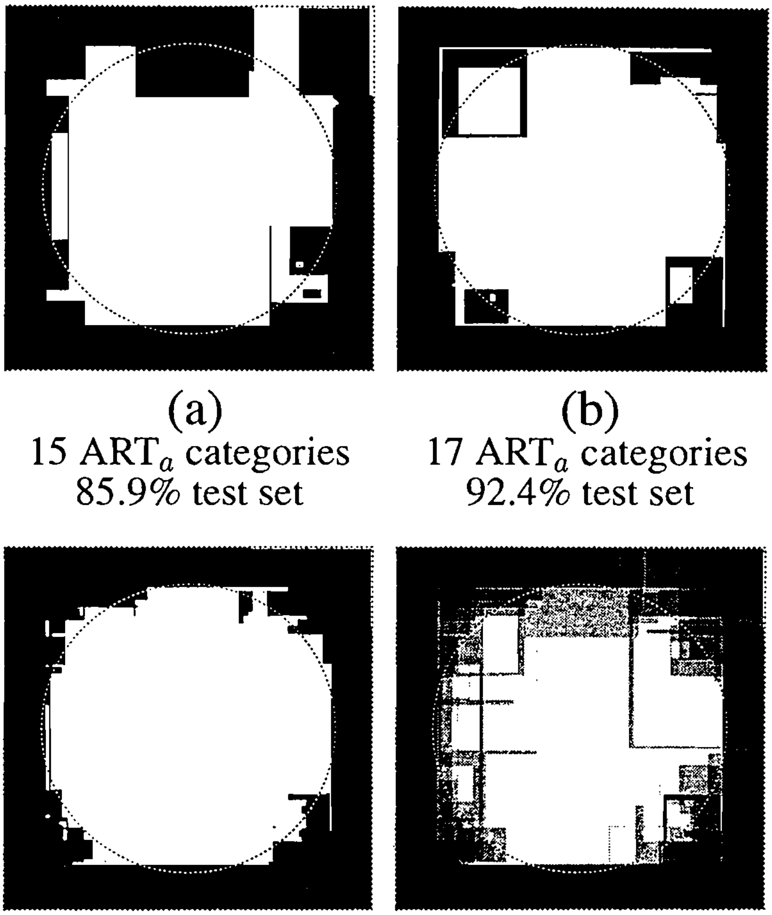
(a)
100 exemplars
99.0% training set
88.6% test set
12 ART$_a$ categories

(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 5. 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. Test set points in white areas are predicted to lie inside the circle and points in black areas are predicted to lie outside the circle. The test set error rate decreases, approximately inversely to the number of ART$_a$ categories, as the training set size increases.



(a)
15 ART$_a$ categories
85.9% test set

(b)
17 ART$_a$ categories
92.4% test set

(c)
Voting on 5 runs
93.9% test set

(d)
Number of votes

Figure 6. 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 used 15 ART$_a$ nodes, test set prediction rate was 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 used 23 ART$_a$ nodes, test set prediction rate was 92.3%, the best of 5 runs. (c) Voting strategy applied to five individual simulations. Test set prediction rate was 93.9%. (d) Cumulative test set response pattern of five 1-epoch simulations. Gray scale intensity increases with the number of votes cast for a point's being outside the circle.

(simplex, duplex, complex, and Gothic) and six different letter styles (block, script, italic, English, Italian, and German)" (p. 162). In addition each image was randomly distorted, leaving many of the characters misshapen. Sixteen numerical feature attributes were then obtained from each character image, and each attribute value was scaled to a range of 0 to 15. The resulting Letter Image Recognition file is archived in the UCI Repository of Machine Learning Databases and Domain Theories, maintained by David Aha and Patrick Murphy (ml_repository@ics.uci.edu).

Frey and Slate used this database to test performance of a family of classifiers based on Holland's genetic algorithms (Holland, 1980). The training set consisted of 16,000 exemplars, with the remaining 4,000 exemplars used for testing. Genetic algorithm classifiers having different input representations, weight update and rule creation schemes, and system parameters were systematically compared. Training was carried out for 5 epochs, plus a sixth "verification" pass during which no new rules were created but a large number of unsatisfactory rules were discarded. In Frey and Slate's comparative study, these systems had correct prediction rates that ranged from 24.5% to 80.8% on the 4,000-item test set. The best performance (80.8%) was obtained using an integer input representation, a reward sharing weight update, an exemplar method of rule creation, and a parameter setting that allowed an unused or erroneous rule to stay in the system for a long time before being discarded. After training, the optimal case, that had 80.8% performance rate, ended with 1,302 rules and 8 attributes per rule, plus over 35,000 more rules that were discarded during verification. (For purposes of comparison, a rule is somewhat analogous to an ART$_a$ category in ARTMAP, and the number of attributes per rule is analogous to the size $|\mathbf{w}_j^a|$ of ART$_a$ category weight vectors.) Building on the results of their comparative study, Frey and Slate investigated two types of alternative algorithms, namely an accuracy-utility bidding system, that had slightly improved performance (81.6%) in the best case; and an exemplar/hybrid rule creation scheme that further improved performance, to a maximum of 82.7%, but that required the creation of over 100,000 rules prior to the verification step.

Fuzzy ARTMAP had an error rate on the letter recognition task that was consistently less than one third that of the three best Frey-Slate genetic algorithm classifiers described above. Moreover Fuzzy ARTMAP simulations each created fewer than 1,070 ART$_a$ categories, compared to the 1,040–1,302 final rules of the three genetic classifiers with the best performance rates. With voting, Fuzzy ARTMAP reduced the error rate to 4.0% (Table 1). Most Fuzzy ARTMAP learning occurred on the first epoch, with test set performance on systems trained for one epoch typically over 97% that of systems exposed to inputs for the five epochs.

Table 1 shows how voting consistently improves performance. In each group, with $\alpha = 0.1$ or $\alpha = 1.0$ and with 1 or 5 training epochs, Fuzzy ARTMAP was run for 3 or 5 independent simulations, each with a different input order. In all cases voting performance was significantly better than performance of any of the individual simulations in a given group. In Table 1a, for example, voting caused the error rate to drop to 8.8%, from a 3-simulation average of 12.5%. With 1 training epoch, 3-simulation voting eliminated about 30–35% of the test set errors (Table 1a and 1c), and 5-simulation voting eliminated about 43% of the test set errors (Table 1e). In the 5-epoch simulations, where individual training set performance was close to 100%, 3-simulation voting still reduced the test set error rate by about 25% (Table 1b and 1d) and 5-simulation voting reduced the error rate by about 34% (Table 1f). The

TABLE 1

	% Correct Test Set Predictions	No. ART_a Categories	No. Epochs
(a) $\alpha = 0.1$			
3 simulations			
Average	87.5%	637	1
Range	87.0%-88.0%	619-661	1
Voting	91.2%		
(b) $\alpha = 0.1$			
3 simulations			
Average	89.7%	741	5
Range	89.3%-90.3%	726-757	5
Voting	92.2%		
(c) $\alpha = 1.0$			
3 simulations			
Average	92.1%	788	1
Range	91.8%-92.3%	762-807	1
Voting	94.8%		
(d) $\alpha = 1.0$			
3 simulations			
Average	94.0%	1,016	5
Range	93.8%-94.3%	988-1,055	5
Voting	95.5%		
(e) $\alpha = 1.0$			
5 simulations			
Average	91.8%	786	1
Range	91.2%-92.6%	763-805	1
Voting	95.3%		
(f) $\alpha = 1.0$			
5 simulations			
Average	93.9%	1,021	5
Range	93.4%-94.6%	990-1,070	5
Voting	96.0%		

Table 1. Voting strategy applied to sets of 3(a–d) or 5(e–f) Fuzzy ARTMAP simulations of the Frey-Slate character recognition task, with choice parameter $\alpha = 0.1$ (a,b) or $\alpha = 1.0$ (c–f); and with training on 1 epoch (a,c,e) or 5 epochs (b,d,f). (a) Voting eliminated 30% of the individual simulation test set errors, which dropped from a 3-simulation average rate of 12.5% to a voting rate of 8.8%. (b) Voting eliminated 24% of the errors, which dropped from 10.3% to 7.8%. (c) Voting eliminated 34% of the errors, which dropped from 7.9% to 5.2%. (d) Voting eliminated 25% of the errors, which dropped from 6.0% to 4.5%. (e) Voting eliminated 43% of the errors, which dropped from 8.2% to 4.7%. (f) Voting eliminated 34% of the errors, which dropped from 6.1% to 4.0%.

best overall results were obtained with $\alpha = 1.0$ and 5-epoch training, where voting reduced the 5-simulation average error rate of 6.1% to a voting error rate of 4.0% (Table 1f).

In summary, single-simulation fast-learn Fuzzy ARTMAP systems, with baseline vigilance $\overline{\rho_a} = 0$ and with choice parameters α ranging from 0.001 to 1.0, were trained on the 16,000-item input set of the Frey-Slate letter recognition task. After 1 to 5 epochs, individual Fuzzy ARTMAP systems had a robust prediction rate of 90% to 94% on the 4,000-item test set, with best performance obtained from the highest values of α. By pooling information across individual simulations, voting consistently eliminated 25%–43% of the errors giving a robust prediction rate of 92%–96%.

REFERENCES

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., 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*, in press.

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.

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.

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.

Frey, P.W. and Slate, D.J. (1991). Letter recognition using Holland-style adaptive classifiers. *Machine Learning*, **6**, 161–182.

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*, **1**, 1–51.

Holland, J.H. (1980). Adaptive algorithms for discovering and using general patterns in growing knowledge bases. *International Journal of Policy Analysis and Information Systems*, **4**, 217-240.

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

Laird, J.E., Newell, A., and Rosenbloom, P.S. (1987). SOAR: An architecture for general intelligence. *Artificial Intelligence*, **33**, 1–64.

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**, pp. 174–185. San Mateo, CA: Morgan Kaufmann Publishers.

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.

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

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

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