Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimensional Maps

Gail A. Carpenter, Stephen Grossberg, Natalya Markuzon, John H. Reynolds, and David B. Rosen, *Student Member, IEEE*

Abstract— **A new neural network architecture is introduced for incremental supervised learning of recognition categories and multidimensional maps in response to arbitrary sequences of analog or binary input vectors, which may represent fuzzy or crisp sets of features. The architecture, called fuzzy ARTMAP, achieves a synthesis of fuzzy logic and adaptive resonance theory (ART) neural networks by exploiting a close formal similarity between the computations of fuzzy subsethood and ART category choice, resonance, and learning. Fuzzy ARTMAP also realizes a new minimax learning rule that conjointly minimizes predictive error and maximizes code compression, or generalization. This is achieved by a match tracking process that increases the ART vigilance parameter by the minimum amount needed to correct a predictive error. As a result, the system automatically learns a minimal number of recognition categories, or "hidden units," to meet accuracy criteria. Category proliferation is prevented by normalizing input vectors at a preprocessing stage. A normalization procedure called complement coding leads to a symmetric theory in which the AND operator ($\vee$) and the OR operator ($\wedge$) of fuzzy logic play complementary roles. Complement coding uses on cells and off cells to represent the input pattern, and preserves individual feature amplitudes while normalizing the total on cell/off cell vector. Learning is stable because all adaptive weights can only decrease in time. Decreasing weights correspond to increasing sizes of category "boxes." Smaller vigilance values lead to larger category boxes. Improved prediction is achieved by training the system several times using different orderings of the input set. This voting strategy can also be used to assign confidence estimates to competing predictions given small, noisy, or incomplete training sets. Four classes of simulations illustrate fuzzy ARTMAP performance in relation to benchmark back-propagation and genetic algorithm systems. These simulations include (i) finding points inside versus outside a circle; (ii) learning to tell two spirals apart, (iii) incremental approximation of a piecewise-continuous function; and (iv) a letter recognition database. The fuzzy ARTMAP system is also compared with Salzberg's NGE system and with Simpson's FMMC system.**

I. Introduction

ARTMAP is a class of neural network architectures that perform incremental supervised learning of recognition categories and multidimensional maps in response to input vectors presented in arbitrary order. The first ARTMAP system [1] was used to classify inputs by the set of features they possess, that is, by an ordered n-tuple (also called a pattern or vector) of binary values representing the presence or absence of each possible feature. This article introduces a more general ARTMAP system that learns to classify inputs by a fuzzy set of features, or a pattern of fuzzy memberships values between 0 and 1 indicating the extent to which each feature is present. This generalization is accomplished by replacing the ART 1 modules [2], [3] of the binary ARTMAP system with fuzzy ART modules [4], [5]. Where ART 1 dynamics are described in terms of set-theory operations, fuzzy ART dynamics are described in terms of fuzzy set-theory operations [6], [7]. Hence the new system is called fuzzy ARTMAP. Also introduced 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 the 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.

Four classes of simulations illustrate fuzzy ARTMAP performance in relation to benchmark back-propagation and genetic algorithm systems. These applications involve analog patterns that are not necessarily interpreted as fuzzy set, but serve to illustrate the properties of the system and allow comparison with several existing 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 and requires no problem-specific system crafting or choice of initial weight values. One way in which fuzzy ARTMAP differs from many previous fuzzy pattern recognition algorithms [8], [9] is that it learns each input as it is received on-line, rather than performing an off-line optimization of a criterion function.

Each ARTMAP system includes a pair of adaptive reso-

Manuscript received July 1, 1991; revised November 1, 1991. This work was supported by British Petroleum (89-A-1204), DARPA (AFOSR 90-0083), the National Science Foundation (NSF IRI 90-00530), the Office of Naval Research (ONR N00014-91-J-4100), and the Air Force Office of Scientific Research (AFOSR 90-0175).

The authors are with the Center for Adaptive Systems and the Department of Cognitive and Neural Systems, Boston University, Boston, MA 02215.

IEEE Log Number 9201469.

1045-9227/92$03.00 © 1992 IEEE

nance theory modules (ART$_a$ and ART$_b$) that create stable recognition categories in response to arbitrary sequences of input patterns (Fig. 1). During supervised learning, ART$_a$ receives a stream $\{\boldsymbol{a}^{(p)}\}$ of input patterns, and ART$_b$ receives a stream $\{\boldsymbol{b}^{(p)}\}$ of input patterns, where $\boldsymbol{b}^{(p)}$ is the correct prediction given $\boldsymbol{a}^{(p)}$. 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 $\boldsymbol{a}^{(p)}$ 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* [1]. 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 $\boldsymbol{a}^{(p)}$ input features that is better able to predict $\boldsymbol{b}^{(p)}$. Because of 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. At the same time, it becomes convenient to switch from the set-theory notation of ART 1 to membership-function or logical notation in which each component of a fixed-length pattern represents whether, or the extent to which, a corresponding feature is present. Thus, the crisp (nonfuzzy) intersection operator ($\cap$) that describes ART 1 dynamics is replaced by the fuzzy AND operator ($\wedge$) of fuzzy set theory [7] in the choice, search, and learning laws of ART 1 (Fig. 2). Especially noteworthy is the close relationship between the computation that defines fuzzy subsethood [6] and the computation that defines category choice in ART 1. Replacing the crisp logical operations of ART 1 with their fuzzy counterparts 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. Because fuzzy ART, and thus fuzzy ARTMAP, can operate on input patterns of continuous values whether or not they represent fuzzy sets, we will often refer to inputs simply as analog patterns or vectors. A neural network realization of the fuzzy ART algorithm is described in [10].

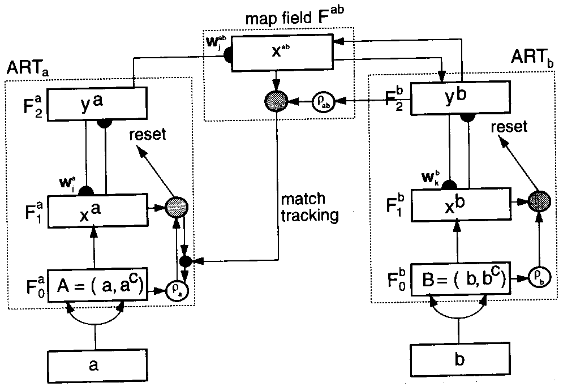


Fig. 1. Fuzzy ARTMAP architecture. The ART$_a$ complement coding preprocessor transforms the M_a vector $\boldsymbol{a}$ into the $2M_a$ vector $\boldsymbol{A} = (\boldsymbol{a}, \boldsymbol{a}^c)$ at the ART$_a$ field $F_0^a \cdot \boldsymbol{A}$ is the input vector to the ART$_a$ field F_1^a. Similarly, the input to F_1^b is the $2M_b$ vector $(\boldsymbol{b}, \boldsymbol{b}^c)$. 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 $|\boldsymbol{x}^a|/|\boldsymbol{A}|$. This triggers an ART$_a$ search which leads to activation of either an ART$_a$ category that correctly predicts $\boldsymbol{b}$ or to a previously uncommitted ART$_a$ category node.

ART 1 (BINARY)	FUZZY ART (ANALOG)
CATEGORY CHOICE	
$T_j = \dfrac{\lvert \mathbf{I} \cap \mathbf{w}_j \rvert}{\alpha + \lvert \mathbf{w}_j \rvert}$	$T_j = \dfrac{\lvert \mathbf{I} \wedge \mathbf{w}_j \rvert}{\alpha + \lvert \mathbf{w}_j \rvert}$
MATCH CRITERION	
$\dfrac{\lvert \mathbf{I} \cap \mathbf{w} \rvert}{\lvert \mathbf{I} \rvert} \geq \rho$	$\dfrac{\lvert \mathbf{I} \wedge \mathbf{w} \rvert}{\lvert \mathbf{I} \rvert} \geq \rho$
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

Fig. 2. Comparison of ART 1 and fuzzy ART.

In fuzzy ART, learning always converges because all adaptive weights are monotonically 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 self-contained summaries of the fuzzy ART and fuzzy ARTMAP systems. Sections II and III describe the fuzzy ART model. Section II summarizes the fuzzy ART equations. Section III describes the model's dynamics, including a geometric interpretation of the fuzzy ART learning process. Section IV contains a comparison between fuzzy ARTMAP, the nested generalized exemplar (NGE) algorithm [11]–[13], and the fuzzy min–max classifier (FMMC) [14]. Section V describes how two fuzzy ART unsupervised learning modules are linked to form the fuzzy ARTMAP supervised learning system.

Sections VI to IX present four classes of benchmark simulation results. Section VI describes a simulation task of learning to identify which points lie inside and which lie outside a given circle. Fuzzy ARTMAP on-line learning (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 two 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 1000-item set for one epoch. These simulations are compared with studies by Wilensky [15] of back-propagation systems. These systems used at least 5000 epochs to reach 90% accuracy on training and testing sets.

Section VII compares fuzzy ARTMAP and back-propagation performance on another benchmark problem, learning to tell two spirals apart. Lang and Witbrock [16] trained a back-propagation system to perform this task in about 20 000 epochs, or in 8000 epochs using the accelerated quickprop weight update rule. With a comparable number of weights, fuzzy ARTMAP performed this task in five epochs. Section VIII shows, by example, how fuzzy ARTMAP creates incremental approximations of arbitrary piecewise-continuous mappings from bounded subsets of $\mathfrak{R}^M$ to bounded subsets of $\mathfrak{R}^N$.

Section IX describes fuzzy ARTMAP performance on a benchmark letter recognition task developed by Frey and Slate [17]. 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. Thus the task is to learn a mapping from $\mathfrak{R}^{16}$ to $\mathfrak{R}^{26}$. Frey and Slate used this database to train a variety of classifiers that incorporate Holland-style genetic algorithms [18]–[20]. Trained on 16 000 exemplars and tested on 4000 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 1070 ART$_a$ recognition categories in all simulations, compared with 1040–1302 rules created by the most accurate genetic algorithms.

II. 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 (Fig. 1). The F_0 activity vector is denoted $\boldsymbol{I} = (I_1, \cdots, I_M)$, with each component I_i in the interval [0,1], $i = 1, \cdots, M$. The F_1 activity vector is denoted $\boldsymbol{x} = (x_1, \cdots, x_M)$ and the F_2 activity vector is denoted $\boldsymbol{y} = (y_1, \cdots, y_N)$. The number of nodes in each field is arbitrary.

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

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

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 monotonically nonincreasing through time and hence converges to a limit. The fuzzy ART weight vector $\boldsymbol{w}_j$ 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 $\boldsymbol{I}$ and F_2 node j, the *choice function*, T_j, is defined by

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

where the fuzzy AND [7] operator $\wedge$ is defined by

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

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

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

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

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

If more than one T_j is maximal, the category j with the smallest index is chosen. In particular, nodes become committed in order $j = 1, 2, 3, \cdots$. When the Jth category is chosen, $y_J = 1$; and $y_j = 0$ for $j \neq J$. In a choice system, the F_1 activity vector $\boldsymbol{x}$ obeys the equation

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

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

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

that is, by (6), when the Jth category is chosen, resonance occurs if

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

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

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

that is, if

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

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 $\boldsymbol{w}_J$ is updated according to the equation

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

Fast learning corresponds to setting $\beta = 1$. The learning law used in the EACH system of Salzberg [11]–[13] is equivalent to (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 $\boldsymbol{w}_J^{(\text{new})} = \boldsymbol{I}$ the first time category J becomes active. Moore [21] 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 certain 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; that is, for some $\gamma > 0$,

$$|\boldsymbol{I}| \equiv \gamma \tag{12}$$

for all inputs $\boldsymbol{I}$. Normalization can be achieved by preprocessing each incoming vector $\boldsymbol{a}$, for example, setting

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

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 $\boldsymbol{a}$ (Fig. 1). To define this operation in its simplest form, let $\boldsymbol{a}$ itself represent the on-response. The complement of $\boldsymbol{a}$, denoted by $\boldsymbol{a}^c$, represents the off-response, where

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

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

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

Note that

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

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

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

III. Fuzzy ART System Dynamics

In fast-learn ART 1, if the choice parameter α in (2) is chosen close to 0, then the first category chosen by (5) will always be the category whose weight vector $\boldsymbol{w}_J$ is the largest coded subset of the input vector $\boldsymbol{I}$, if such a category exists [2]. As shown below, the choice function T_j in (2) can then be interpreted as a fuzzy membership of the input $\boldsymbol{I}$ in the jth category. Then $\boldsymbol{w}_J$ is chosen if $w_{Ji} = 0$ where $I_i = 0$, and if it has the maximal number of 1's ($w_{Ji} = 1$) at indices i where $I_i = 1$, among all weight vectors $\boldsymbol{w}_j$. Moreover, when $\boldsymbol{w}_J$ is a subset of $\boldsymbol{I}$ during resonance, $\boldsymbol{w}_J$ is unchanged, or conserved, during learning. The limit $\alpha \to 0$ is called the *conservative limit* because small values of α tend to minimize recoding during learning. Note that in simulations, α must be large enough to affect the values of the choice function T_j in (2) after round-off. In all simulations described in this article, $\alpha \geq 0.001$.

For analog vectors, the degree to which $\boldsymbol{q}$ is a fuzzy subset of $\boldsymbol{p}$ is given by the term

$$\frac{|\boldsymbol{p} \wedge \boldsymbol{q}|}{|\boldsymbol{q}|} \tag{18}$$

[6]. In the conservative limit of fuzzy ART, the choice function T_j in (2) primarily reflects the degree to which the weight vector $\boldsymbol{w}_J$ is a fuzzy subset of the input vector $\boldsymbol{I}$. If

$$\frac{|\boldsymbol{I} \wedge \boldsymbol{w}_j|}{|\boldsymbol{w}_j|} = 1, \tag{19}$$

then $\boldsymbol{w}_j$ is a fuzzy subset of $\boldsymbol{I}$ [7] and category j is said to be a *fuzzy subset choice* for input $\boldsymbol{I}$. When a fuzzy subset category choice exists, it is always selected over other choices. In this case, by (11), no recoding occurs if j is selected since $\boldsymbol{I} \wedge \boldsymbol{w}_j = \boldsymbol{w}_j$. If more than one category is a fuzzy subset choice, the small but positive parameter α breaks the tie by choosing J that maximizes $|\boldsymbol{w}_j|$ among the fuzzy subset choices.

Resonance depends on the degree to which $\boldsymbol{I}$ is a fuzzy subset of $\boldsymbol{w}_J$, by (7) and (9). The close linkage between fuzzy subsethood and ART choice/resonance/learning forms the foundation of the computational properties of fuzzy ART. In particular, if category j is a fuzzy subset choice, then the match function value is given by

$$\frac{|\boldsymbol{I} \wedge \boldsymbol{w}_j|}{|\boldsymbol{I}|} = \frac{|\boldsymbol{w}_j|}{|\boldsymbol{I}|}. \tag{20}$$

Thus, choosing J to maximize $|\boldsymbol{w}_j|$ among fuzzy subset choices also maximizes the opportunity for resonance in (7). If reset occurs for the node that maximizes $|\boldsymbol{w}_j|$, then reset will also occur for all other subset choices. In the conservative limit ($\alpha \cong 0$) with fast learning ($\beta = 1$) and normalized inputs, one-shot stable learning occurs; that is, no weight change or search occurs in a fuzzy ART system after each item of an input set is presented *just once,* although some inputs may select different categories on future trials [5]. The one-shot learning property holds for fuzzy ART modules with constant vigilance. In fuzzy ARTMAP, where vigilance can vary when predictive errors are made, repeated input presentations can lead to new learning, so one-shot learning does not necessarily occur.

A geometric interpretation of fuzzy ART with complement coding will now be outlined. For definiteness, let the input set consist of two-dimensional vectors $\boldsymbol{a}$ preprocessed into the four-dimensional complement coding form. Thus

$$\boldsymbol{I} = (\boldsymbol{a}, \boldsymbol{a}^c) = (a_1, a_2, 1 - a_1, 1 - a_2). \tag{21}$$

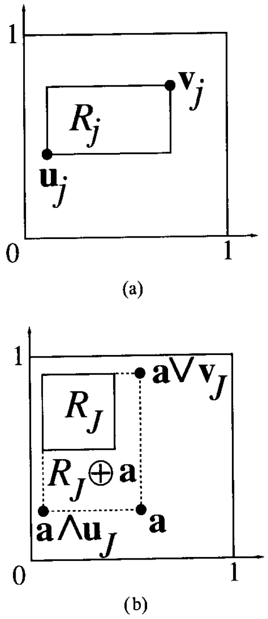
In this case, each category j has a geometric representation as a rectangle R_j, as follows. Following (21), the weight vector $\boldsymbol{w}_j$ can be written in complement coding form:

$$\boldsymbol{w}_j = \left(\boldsymbol{u}_j, \boldsymbol{v}_j^c\right), \tag{22}$$

where $\boldsymbol{u}_j$ and $\boldsymbol{v}_j$ are two-dimensional vectors. Let vector $\boldsymbol{u}_j$ define one corner of a rectangle R_j and let $\boldsymbol{v}_j$ define another corner of R_j (Fig. 3(a)). The size of R_j is defined to be

$$|R_j| \equiv |\boldsymbol{v}_j - \boldsymbol{u}_j| \tag{23}$$

which is equal to the height plus the width of R_j in Fig. 3(a).



(a)

(b)

Fig. 3. Fuzzy ART weight representation. (a) In complement coding form with $M = 2$, each weight vector $\boldsymbol{w}_j$ has a geometric interpretation as a rectangle R_j with corners $(\boldsymbol{u}_j, \boldsymbol{v}_j)$. (b) During fast learning, R_j expands to $R_j \oplus \boldsymbol{a}$, the smallest rectangle that includes R_j and $\boldsymbol{a}$, provided that $|R_j \oplus \boldsymbol{a}| \leq 2(1 - \rho)$.

In a fast-learn fuzzy ART system, with $\beta = 1$ in (11), $\boldsymbol{w}_J^{(\text{new})} = \boldsymbol{I} = (\boldsymbol{a}, \boldsymbol{a}^c)$ when J is an uncommitted node. The corners of $R_J^{(\text{new})}$ are then given by $\boldsymbol{a}$ and $(\boldsymbol{a}^c)^c = \boldsymbol{a}$. Hence $R_J^{(\text{new})}$ is just the point $\boldsymbol{a}$. Learning increases the size of each R_j. In fact the size of R_j grows as the size of $\boldsymbol{w}_j$ shrinks during learning, and the maximum size of R_j is determined by the vigilance parameter ρ, as shown below. During each fast-learning trial, R_J expands to $R_J \oplus \boldsymbol{a}$, the minimum rectangle containing R_J and $\boldsymbol{a}$ (Fig. 3(b)). The corners of $R_J \oplus \boldsymbol{a}$ are given by $\boldsymbol{a} \wedge \boldsymbol{u}_j$ and $\boldsymbol{a} \vee \boldsymbol{v}_J$, where the fuzzy AND operator, $\wedge$, is defined by (3); and the fuzzy OR operator, $\vee$, is defined by

$$(\boldsymbol{p} \vee \boldsymbol{q})_i \equiv \max(p_i, q_i) \tag{24}$$

[7]. Hence, by (23), the size of $R_J \oplus \boldsymbol{a}$ is given by

$$|R_J \oplus \boldsymbol{a}| \equiv |(\boldsymbol{a} \vee \boldsymbol{v}_J) - (\boldsymbol{a} \wedge \boldsymbol{u}_J)|. \tag{25}$$

However, reset leads to another category choice if $|R_J \oplus \boldsymbol{a}|$ is too large. In summary, with fast learning, each R_j equals the smallest rectangle that encloses all vectors $\boldsymbol{a}$ that have chosen category j, under the constraint that $|R_j| \leq 2(1 - \rho)$.

In general, if $\boldsymbol{a}$ has dimension M, the hyperrectangle R_j includes the two vertices $\wedge_j \boldsymbol{a}$ and $\vee_j \boldsymbol{a}$, where the ith component of each vector is defined by the equations

$$(\wedge_j \boldsymbol{a})_i = \min\{a_i : \boldsymbol{a} \text{ has been coded by category } j\} \tag{26}$$

and

$$(\vee_j \boldsymbol{a})_i = \max\{a_i : \boldsymbol{a}\text{has been coded by category } j\} \tag{27}$$

(Fig. 4). The size of R_j is given by

$$|R_j| = |\vee_j \boldsymbol{a} - \wedge_j \boldsymbol{a}| \tag{28}$$

and the weight $\boldsymbol{w}_j$ is given by

$$\boldsymbol{w}_j = (\wedge_j \boldsymbol{a}, (\vee_j \boldsymbol{a})^c), \tag{29}$$

as in (22) and (23). Thus

$$|\boldsymbol{w}_j| = \sum_i (\wedge_j \boldsymbol{a})_i + \sum_i [1 - (\vee_j \boldsymbol{a})_i] = M - |\vee_j \boldsymbol{a} - \wedge_j \boldsymbol{a}|, \tag{30}$$

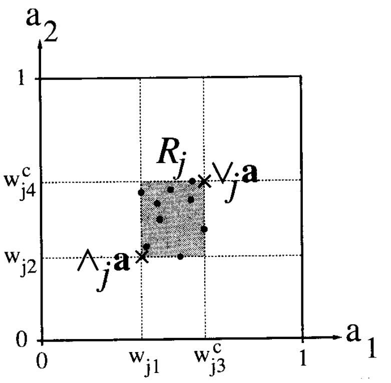


Fig. 4. With fuzzy ART fast learning and complement coding, the jth category rectangle R_j includes all those vectors $\boldsymbol{a}$ in the unit square which have activated category j without reset. The weight vector $\boldsymbol{w}_j$ equals $(\wedge_j\boldsymbol{a}, (\vee_j\boldsymbol{a})^c)$.

and the size of the hyperrectangle R_j is therefore

$$|R_j| = M - |\boldsymbol{w}_j|. \tag{31}$$

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

$$|\boldsymbol{w}_j| \geq \rho M. \tag{32}$$

By (31) and (32),

$$|R_j| \leq (1-\rho)M. \tag{33}$$

Thus 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, $|\boldsymbol{w}_j| = 2M$, by (17), and so $|R_j| \equiv -M$, by (31). These observations may be combined into the following theorem [5].

Theorem: Fuzzy ART Stable Category Learning: A fuzzy ART system with complement coding, fast learning, and constant vigilance forms hyperrectangular categories that converge to limits in response to an arbitrary sequence of analog or binary input vectors. Hyperrectangles grow monotonically in all dimensions. The size $|R_j|$ of a hyperrectangle equals $M - |\boldsymbol{w}_j|$, where $\boldsymbol{w}_j$ 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.

IV. A Comparison of Fuzzy ARTMAP, NGE, and FMMC

The geometry of the hyperrectangles R_j resembles parts of the nested generalized exemplar (NGE) system [11]–[13] and the fuzzy min–max classifier (FMMC) system [14]. Both NGE and FMMC, as well as fuzzy ARTMAP, use hyperrectangles 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 significantly, however, in their responses to incorrect predictions. In particular, because NGE and FMMC do not have components that play the role of the ART vigilance parameter, these algorithms do not have the property of internal control of hyperrectangle 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 hyperrectangles 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 learns only the first exemplar of each category, is necessarily stable. Using stability and match tracking, fuzzy ARTMAP automatically constructs as many categories as are needed to learn any consistent training set to 100% accuracy. Both ARTMAP and NGE rely on multilayer structures to effect their learning strategies. In contrast, the FMMC is a two-layer, feedforward system. In particular, each output class is associated with only one category box. It can therefore learn only a very limited set of possible category structures. In contrast, fuzzy ARTMAP can learn to associate multiple categories with the same output, as would be needed to name capital letters, script letters, and other letter fonts with the same letter name or, more generally, multiple disconnected clusters of features with the same output classification.

V. 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 that 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 [1]. 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$_a$, $\boldsymbol{I} = \boldsymbol{A} = (\boldsymbol{a}, \boldsymbol{a}^c)$; and for ART$_b$, $\boldsymbol{I} = \boldsymbol{B} = (\boldsymbol{b}, \boldsymbol{b}^c)$ (Fig. 1). Variables in ART$_a$ or ART$_b$ are designated by subscripts or superscripts a and b. For ART$_a$, let $\boldsymbol{x}^a \equiv (x_1^a \cdots x_{2M_a}^a)$ denote the F_1^a output vector; let $\boldsymbol{y}^a \equiv (y_1^a \cdots y_{N_a}^a)$ denote the F_2^a output vector; and let $\boldsymbol{w}_j^a \equiv (w_{j1}^a, w_{j2}^a, \cdots, w_{j,2M_a})$ denote the jth ART$_a$ weight vector. For ART$_b$, let $\boldsymbol{x}^b \equiv (x_1^b \cdots x_{2M_b}^b)$ denote the F_1^b output vector; let $\boldsymbol{y}^b \equiv (y_1^b \cdots y_{N_b}^b)$ denote the F_2^b output vector; and let $\boldsymbol{w}_k^b \equiv (w_{k1}^b, w_{k2}^b, \cdots, w_{k,2M_b}^b)$ denote the kth ART$_b$ weight vector. For the map field, let $\boldsymbol{x}^{ab} \equiv (x_1^{ab}, \cdots, x_{N_b}^{ab})$ denote the F_{ab} output vector, and let $\boldsymbol{w}_j^{ab} \equiv (w_{j1}^{ab}, \cdots, w_{jN_b}^{ab})$ denote the weight vector from the jth F_2^a node to F^{ab}. Vectors $\boldsymbol{x}^a$, $\boldsymbol{y}^a$, $\boldsymbol{x}^b$, $\boldsymbol{y}^b$, and $\boldsymbol{x}^{ab}$ are set to $\boldsymbol{0}$ 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 $\boldsymbol{w}_J^{ab}$ 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 $\boldsymbol{w}_J^{ab}$. The F^{ab} output vector $\boldsymbol{x}^{ab}$ obeys

$$\boldsymbol{x}^{ab} = \begin{cases} \boldsymbol{y}^b \wedge \boldsymbol{w}_J^{ab} & \text{if the } J\text{th } F_2^a \text{ node is active and } F_2^b \text{ is active} \\ \boldsymbol{w}_J^{ab} & \text{if the } J\text{th } F_2^a \text{ node is active and } F_2^b \text{ is inactive} \\ \boldsymbol{y}^b & \text{if } F_2^a \text{ is inactive and } F_2^b \text{ is active} \\ \mathbf{0} & \text{if } F_2^a \text{ is inactive and } F_2^b \text{ is inactive.} \end{cases} \tag{34}$$

By (34), $\boldsymbol{x}^{ab} = \mathbf{0}$ if the prediction $\boldsymbol{w}_J^{ab}$ is disconfirmed by $\boldsymbol{y}^b$. 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

$$|\boldsymbol{x}^{ab}| < \rho_{ab}|\boldsymbol{y}^b|, \tag{35}$$

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

$$|\boldsymbol{x}^a| = |\boldsymbol{A} \wedge \boldsymbol{w}_J^a| < \rho_a|\boldsymbol{A}|, \tag{36}$$

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

$$|\boldsymbol{x}^a| = |\boldsymbol{A} \wedge \boldsymbol{w}_J^a| \geq \rho_a|\boldsymbol{A}| \tag{37}$$

and

$$|\boldsymbol{x}^{ab}| = |\boldsymbol{y}^b \wedge \boldsymbol{w}_J^{ab}| \geq \rho_{ab}|\boldsymbol{y}^b|; \tag{38}$$

or, if no such node exists, to the shutdown 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 $F_2^a \rightarrow F^{ab}$ paths initially satisfy

$$w_{jk}^{ab}(0) = 1. \tag{39}$$

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

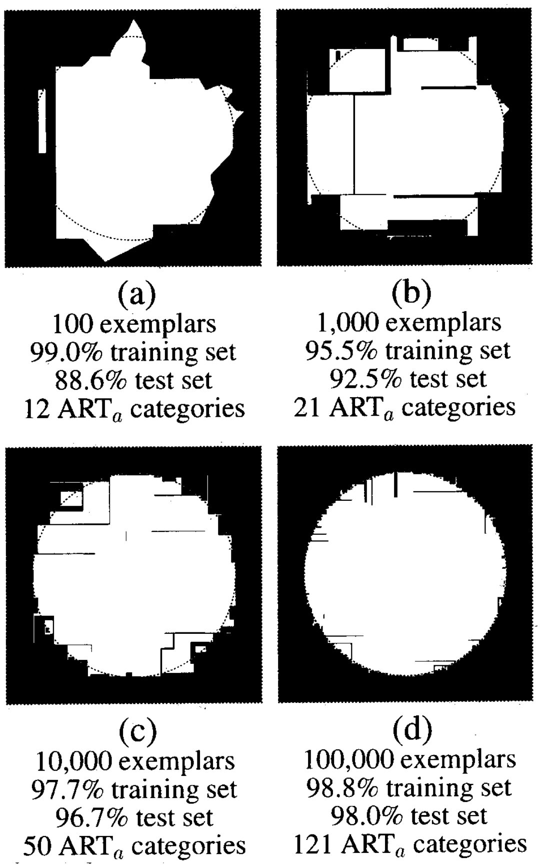
VI. Simulation: Circle-in-the-Square

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 [15]. 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 overfitting, 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 Figs. 5 and 6. As training set size increased from 100 exemplars (Fig. 5(a)) to 100 000 exemplars (Fig. 5(d)) 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 $\boldsymbol{w}_j^a$ in ART$_a$ plus one map field weight $\boldsymbol{w}_j$ to record whether category j predicts that a point lies inside or outside the circle. Thus, for example, one-epoch training on 100 exemplars used 60 weights to achieve 88.6% test set accuracy. Fig. 6 shows the ART$_a$ category rectangles R_j^a established in each simulation of Fig. 5. Initially, large R_j^a estimated large areas as belonging to one or the other category (Fig. 6(a)). Additional R_j^a's improved accuracy, especially near the boundary of the circle (Fig. 6(d)). The map can be made arbitrarily accurate provided the number of ART$_a$ nodes is allowed to increase as needed. As in Fig. 3 each rectangle R_j^a corresponds to the four dimensional weight vector $\boldsymbol{w}_j^a = (\boldsymbol{u}_j^a, (\boldsymbol{v}_j^a)^c)$, where $\boldsymbol{u}_j^a$ and $\boldsymbol{v}_j^a$ are plotted as the lower-left and upper-right corners of R_j^a, respectively.

Fig. 7 depicts the response patterns of fuzzy ARTMAP on another series of circle-in-the-square simulations that use the same training sets as in Fig. 5. However, each input set was presented for as many epochs as were needed to achieve 100% predictive accuracy on the training set, whereas in Fig. 5 each training set was presented for only one epoch. In each case, test set predictive accuracy increased, as did the number of ART$_a$ category nodes. For example, with 10 000 exemplars, one epoch training used 50 ART$_a$ nodes to give 96.7% test set accuracy (Fig. 5(c)). The same training set, after six epochs, used 89 ART$_a$ nodes to give 98.3% test set accuracy (Fig. 7(c)).

Fig. 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 one epoch simulations. Fig. 7 shows how a 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.



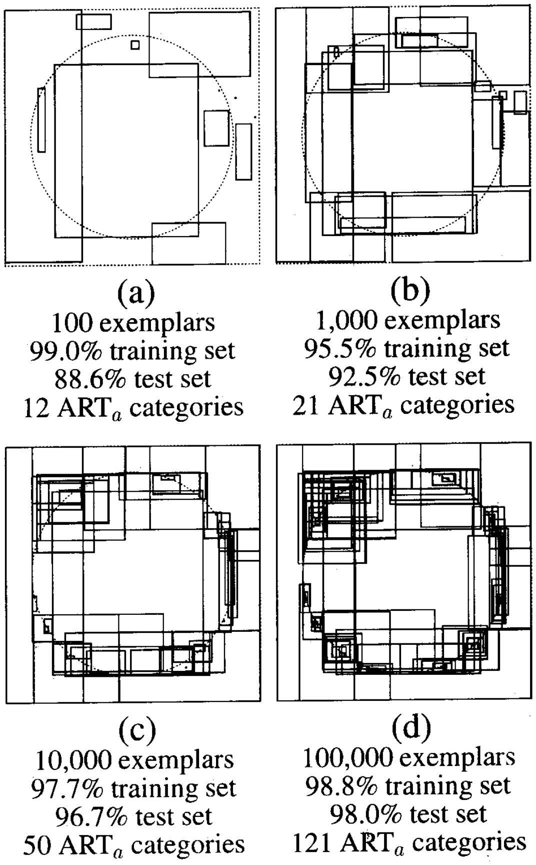
(a)
100 exemplars
99.0% training set
88.6% test set
12 ART$_a$ categories

(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

Fig. 5. Circle-in-the-square test set response patterns after one epoch of fuzzy ARTMAP training on (a) 100, (b) 1000, (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)
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

Fig. 6. Fuzzy ARTMAP category rectangles R_j^a for the circle-in-the-square simulations of Fig. 5. Small rectangles are created near the map discontinuities as the error rate drops toward 0.

A fixed set of 1000 randomly chosen exemplars was presented to a fuzzy ARTMAP system on five independent one-epoch circle-in-the-square simulations. After each simulation, inside/outside predictions were recorded on a 1000-item test set. Accuracy on individual simulations ranged from 85.9% to 93.4%, 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% (Fig. 8(c)). In other words, test set errors were reduced from an average individual rate of 9.5% to a voting rate of 6.1%. Fig. 8(d) 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 1000-item training set. By comparison, a tenfold 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 Fig. 5(b), one-epoch training on 1000 items yielded at 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% (Fig. 5(c)).

In the circle-in-the-square simulations, $M_a = 2$, and ART$_a$ inputs $\boldsymbol{a}$ were randomly chosen points in the unit square. Each F_1^a input $\boldsymbol{A}$ had the form

$$\boldsymbol{A} = (a_1, a_2, 1 - a_1, 1 - a_2), \tag{40}$$

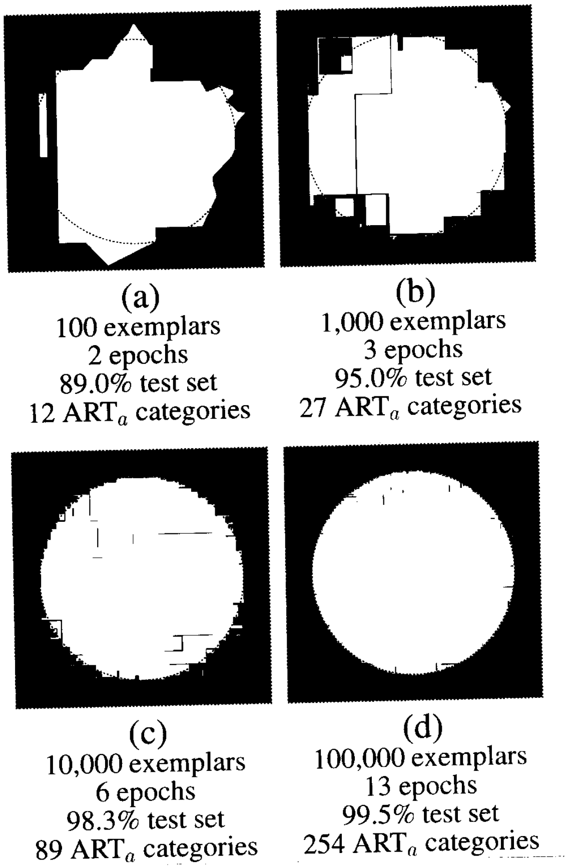
and $|\boldsymbol{A}| = 2$. For ART$_b$, $M_b = 1$. The ART$_b$ input $\boldsymbol{b}$ was given by

$$\boldsymbol{b} = \begin{cases} (1) & \text{if } \boldsymbol{a} \text{ is inside the circle} \\ (0) & \text{otherwise.} \end{cases} \tag{41}$$

In complement coding form, the $F_0^b \to F_1^b$ input $\boldsymbol{B}$ is given by

$$\boldsymbol{B} = \begin{cases} (1,0) & \text{if } \boldsymbol{a} \text{ is inside the circle} \\ (0,1) & \text{otherwise.} \end{cases} \tag{42}$$

The fuzzy ARTMAP simulations used fast learning, defined by (11) with $\beta = 1$; the choice parameter $\alpha \cong 0$ in the conservative limit for both ART$_a$ and ART$_b$; and the baseline vigilance parameter $\overline{\rho_a} = 0$. The vigilance parameters ρ_{ab} and ρ_b can be set to any value between 0 and 1 without affecting fast-learn results. In each simulation, the system was trained on the specified number of exemplars, then tested on 1000 or more points.



(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

Fig. 7. Circle-in-the-square test set response patterns with exemplars repeatedly presented until the system achieved 100% correct prediction on (a) 100, (b) 1000 (c) 10 000, and (d) 100 000 training set points. Training sets were the same as those used for Figs. 5 and 6. Training to 100% accuracy required (a) two epochs, (b) three epochs, (c) six epochs, and (d) 13 epochs. Additional training epochs decreased test set error rates created additional ART$_a$ categories, compared with the one epoch simulation in Fig. 5.

VII. Simulation: Learning to Tell Two Spirals Apart

Learning to tell two spirals apart is a neural network benchmark task proposed by A.P. Wieland [16]. The two spirals of the benchmark task each make three complete turns in the plane, with 32 points per turn plus an endpoint, totaling 97. During one epoch, the outermost white point is presented first, then the outermost black point, and so on, working in to the center of each spiral. Specifically, the training set exemplar sequence $(\boldsymbol{a}^{(1)},\boldsymbol{b}^{(1)}),(\boldsymbol{a}^{(2)},\boldsymbol{b}^{(2)})\cdots$, with $\boldsymbol{a}^{(t)} = \left(a_1^{(t)},a_2^{(t)}\right) \in \Re^2$ and $\boldsymbol{b}^{(t)} = (b^{(t)}) \in \Re^1$, is given by the following equations. For $n = 1,2,\cdots,97$,

$$a_1^{(2n-1)} = 1 - a_1^{(2n)} = r_n \sin\alpha_n + 0.5 \qquad (43)$$

$$a_1^{(2n-1)} = 1 - a_2^{(2n)} = r_n \cos\alpha_n + 0.5 \qquad (44)$$

where

$$r_n = 0.4\left(\frac{105-n}{104}\right), \qquad (45)$$

$$\alpha_n = \frac{\pi(n-1)}{16}, \qquad (46)$$

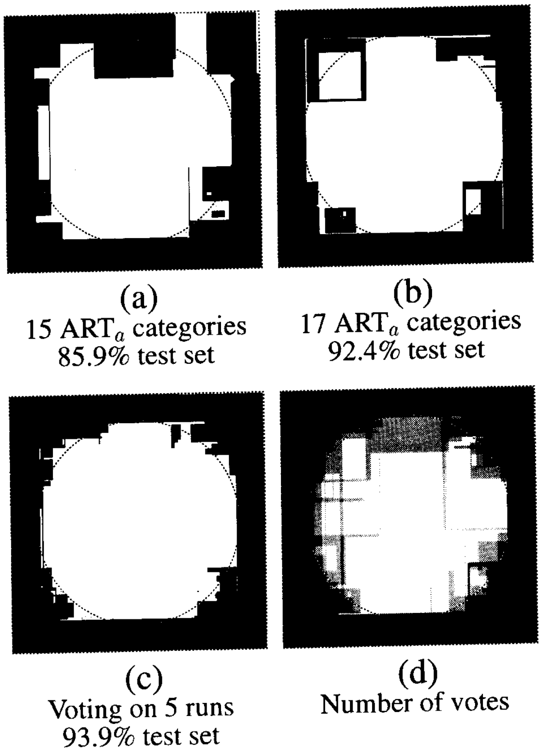
$$b^{(2n-1)} = 1 \quad \text{[white]}, \qquad (47)$$

and

$$b^{(2n)} = 0 \quad \text{[black]}. \qquad (48)$$

Lang and Witbrock [16] constructed a back-propagation system that learned to distinguish points on the two intertwined spirals. They reported being unable to accomplish this task using a standard back-propagation system, with connections from one layer to the next. They then crafted a special 2–5–5–5–1 system with shortcut connections, each node being connected to all nodes in all subsequent layers. With one additional bias weight for each node, therefore, the system had 138 trainable weights.

Lang and Witbrock considered the task to be complete when each of the 194 points in the training set responded to within 0.4 of its target output value, equal to 0 on the black spiral, 1 on the white spiral. Training time was measured in epochs, the number of times the entire training set was presented.



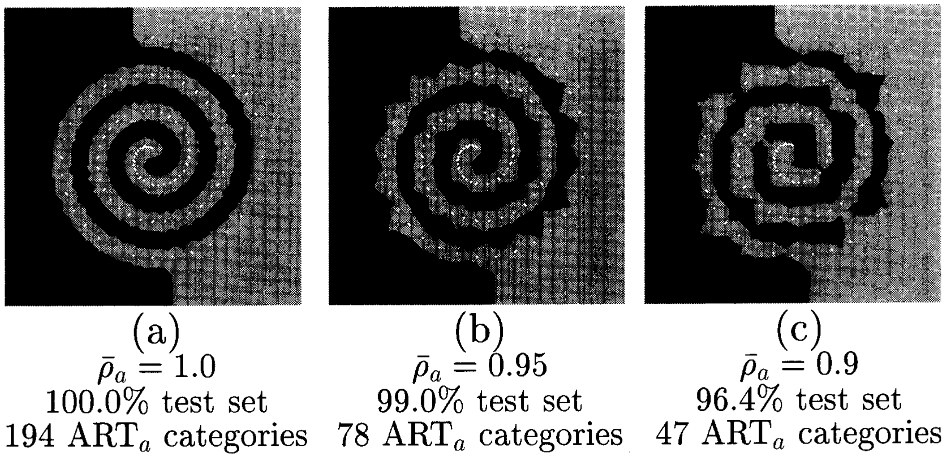
(a)
15 ART$_a$ categories
85.9% test set

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

(c)
Voting on 5 runs
93.9% test set

(d)
Number of votes

Fig. 8. Circle-in-the-square response patterns for a fixed 1000–item training set. (a) Test set responses after training on inputs presented in random order. After one epoch that used 15 ART$_a$ nodes, test set prediction rate was 85.9%, the worst of five runs. (b) Test set responses after training on inputs presented in a different random order. After one epoch that used 17 ART$_a$ nodes, test set prediction rate was 92.4%, the best of five runs. (c) Voting strategy applied to five individual simulations. Test set prediction rate was 93.9%. (d) Cumulative test set response pattern of five one-epoch simulations. Gray scale intensity increases with the number of votes cast for a point's being outside the circle.



(a) $\bar{\rho}_a = 1.0$ 100.0% test set 194 ART_a categories

(b) $\bar{\rho}_a = 0.95$ 99.0% test set 78 ART_a categories

(c) $\bar{\rho}_a = 0.9$ 96.4% test set 47 ART_a categories

Fig. 9. Intertwined black and white spiral training points. Each spiral has 97 points and three turns. Plots show unit square response patterns after one training epoch. Lighter areas predict white spiral points and darker areas predict black spiral points. The 194 training set points are shown superimposed on the response patterns. (a) Fuzzy ARTMAP creates a nearest neighbor classifier in one epoch when $\overline{\rho_a} = 1$. After one epoch, classification is 100% correct on both the training set spirals (194 points) and on the test set dense spirals (770 points). (b) With $\overline{\rho_a} = 0.95$, 78 ART_a categories are created. Accuracy is 99.0% on the dense spiral. (c) With $\overline{\rho_a} = 0.9$, 47 ART_a categories are created. Accuracy is 96.4% on the dense spiral.

When weights were updated using vanilla back-propagation [22], training required an average of 20 000 epochs. Average training time was decreased to 10 000 epochs using the cross-entropy error measure and to 8000 epochs using the quickprop algorithm [23].

Fig. 9(a) shows the two-spiral response pattern of a fuzzy ARTMAP simulation with baseline vigilance $\overline{\rho}_a = 1.0$. Points in the light gray areas predict white, while points in the dark gray areas predict black. In just 1 epoch, each input established its own category, creating a nearest neighbor classifier that is 100% correct on the 194-point training set. Moreover, 100% correct prediction is achieved on a test set consisting of two dense spirals, each with 385 points. However, the necessary number of stored weights is 582, two to assign each point $(a_1^{(t)}, a_2^{(t)})$ an ART_a category index $j\,(j = 1, \cdots, 194)$; and one to label that category black or white. The main goal of both the back-propagation and fuzzy ARTMAP simulations described below is to reduce the number of weights needed for this task. Progress toward this goal was made by reducing the baseline vigilance parameter $\overline{\rho_a}$ to 0.95, which reduced the number of ART_a categories to 78 (Fig. 9(b)), less than 40% the number previously needed. On the dense spiral test set, accuracy was still 99.0%. Note that, since $\rho_a \geq \overline{\rho_a}$, with $\overline{\rho_a} = 0.95$ the maximum size of a category rectangle R_j^a was $2(1 - \overline{\rho_a}) = 0.1$, by (33). Reducing $\overline{\rho_a}$ to 0.9 (Fig. 9(c)) allowed rectangles to grow to a maximum size of 0.2. This reduced the number of ART_a nodes to 47. Performance on the dense spiral set also dropped to 96.4%, with the total response pattern showing some gaps in the spiral pattern.

When $\overline{\rho_a}$ was reduced to 0, fuzzy ARTMAP learned to tell the two spirals apart in five epochs, using 25 ART_a categories (125 weights) compared with 138 weights in the Lang–Witbrock system. Table I shows the number of epochs needed to reach various intermediate stages of training for fuzzy ARTMAP, with $\overline{\rho_a} = 0$; and for the Lang–Witbrock system, with vanilla back-propagation and quickprop training.

TABLE I
NUMBER OF TRAINING EPOCHS IN TWO-SPIRAL SIMULATIONS

Cases Remaining to Learn (194 total)	Fuzzy ARTMAP (125 weights) $\overline{\rho_a} = 0$	Lang–Witbrock Back- Propagation (138 weights)			
		Vanilla BP		Quickprop	
		A	C	A	C
18	1	9000	16 800	2100	6600
11	2	13 000	17 400	2500	6700
0	5	18 900	19 000	4500	6800

* The number of epochs for intermediate stages of vanilla back-propagation and quickprop training have been estimated from Lang and Witbrock [16, Figs. 8 and 10]. Runs A and C correspond to distinct sets of random initial weights. On average, vanilla back-propagation required 20 000 epochs and quickprop required 8000 epochs for the system to learn to distinguish points in the training set. See Fig.10.

The first column gives the number of cases remaining to learn out of the 194 training set items. Fuzzy ARTMAP learned all but 18 cases in one epoch, and corrected these errors by the fifth epoch. For the Lang–Witbrock black-propagation simulations, runs A and C corresponded to two different sets of initial weights. With both vanilla back-propagation and quickprop learning rules, run A was more accurate early in training, but needed many epochs to learn the last few cases. Run C, in contrast, tended to have long intervals where little improvement occurred, but converged rapidly at the end. Of the two, run C showed better generalization. For example, on the dense spiral test set, run A (vanilla back-propagation) placed 90.2% of the points on the correct spiral, while run C placed 92.8% on the correct spiral.

Fig. 10 illustrates the fuzzy ARTMAP response patterns after one, two, and five training epochs. During the first epoch, the system generated two intertwined square wave spirals (Fig. 10(a)). After one epoch, correct predictions were already made by 91% of the points on the training set, with errors

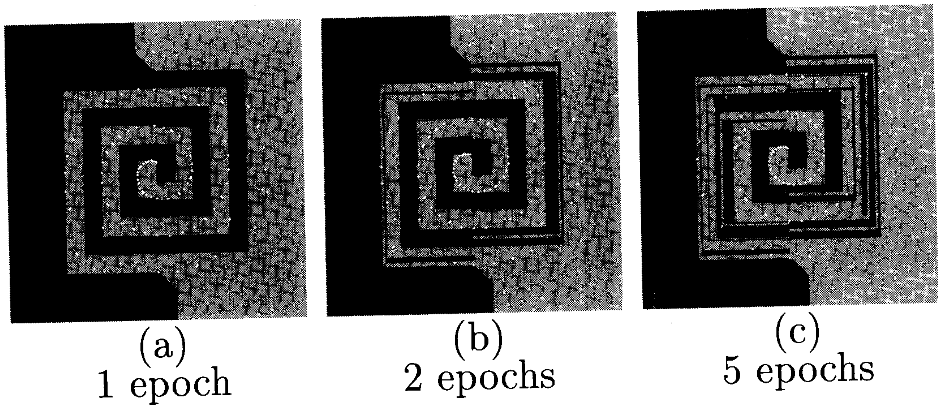


Fig. 10. Fuzzy ARTMAP response patterns during a two-spiral simulation with $\overline{\rho_a} = 0.0$. (a) After one epoch, the system has learned correct responses for 176 training set points (90.7%). Errors are concentrated in the outer corners of the square spirals. The system has established 13 ART$_a$ categories, which requires 65 weights. (b) After two epochs, the system has established 16 ART$_a$ categories, giving 183 correct responses (94.3%) in the training set. (c) After five epochs, the task is complete, with correct responses at all 194 training set points, and 25 ART$_a$ categories. See Table I.

occurring in the corners of the outer turns of the square spirals. Correct predictions were also made on 86% of points in the dense spiral test set. On subsequent epochs, fuzzy ARTMAP quickly corrected all errors in the training set (Fig. 10(c)). The additional training did not improve prediction on the dense spiral test set, however. Inspection of Fig. 10 suggests that overfitting may, in fact, be occurring. Overfitting could also occur with back-propagation if the identification criterion were made stricter than the 0.4 tolerance used by Lang and Witbrock. In Fuzzy ARTMAP, a similar softer criterion could be used with slow learning in the map field weight vector $\boldsymbol{w}_j$, rather than the extreme case of fast learning in all weights. Alternatively, a voting strategy on simulations that scramble input order could also be used.

VIII. Simulation: Incremental Approximation of a Piecewise-Continuous Function

The spiral and circle-in-the square tasks require a system to learn a piecewise-continuous map from $\Re^2$ to $\Re^1$. This is a special case of the task of learning to approximate a piecewise-continuous function from $\Re^n$ to $\Re^m$. In this section we illustrate how fuzzy ARTMAP builds an approximate representation of a continuous sinusoidal function from $\Re^1$ to $\Re^1$, namely

$$b = f(a) = (\sin 2\pi a)^2 \tag{49}$$

for $0 \leq a \leq 1$. The approximation is improved incrementally as each new data point is added. The asymptotic accuracy of the approximation is determined by the ART$_b$ vigilance parameter, ρ_b. For ρ_b close to 1, the approximation reaches an arbitrary degree of accuracy, given sufficiently many ART$_a$ and ART$_b$ category nodes.

Fig. 11 shows how the function approximation evolves as data points are received. Since the dimension (M_a) of ART$_a$ inputs and the dimension (M_b) of ART$_b$ inputs are both equal to 1, fuzzy ARTMAP establishes category intervals R^a_j and R^b_k during learning. Fig. 11 indicates the evolution of these category intervals as the number of training inputs increased from (a) 10 to (b) 25 to (c) 50. The ART$_b$ vigilance parameter ρ_b was set equal to 0.9. By (33), therefore, the maximum length of an R^b_k interval was $0.1 = (1 - \rho_b)$.

To interpret the geometry of Fig. 11, consider the following example. If an input $\boldsymbol{a} = (a)$ chooses a committed ART$_a$ category J, and if a lies in the interval R^a_J, then fuzzy ARTMAP predicts that b lies in the interval R^b_K for some K. If b is in R^b_K, no new learning occurs. If b does not lie in R^b_K, but the size of the expanded interval $R^b_K \oplus \boldsymbol{b}$ is less than 0.1, then R^b_K will expand to include the new point b. If, on the other hand, the size of $R^b_K \oplus \boldsymbol{b}$ exceeds the maximum length (0.1) established by ρ_b, then another ART$_b$ node (K') is selected. Inter-ART reset and match tracking then lead to the selection of another (possibly uncommitted) ART$_a$ category. The ARTMAP search process continues until a selected ART$_a$ category node J' correctly predicts the ART$_b$ category K', or until an uncommitted ART$_a$ node becomes active. In the latter case, an ART$_a$ point interval $R^a_{J'} = [a, a]$ learns to predict the ART$_b$ interval $R^b_{K'}$, and a new ART$_a$ category is established. In either case, during learning, components of the map field weight vector $\boldsymbol{w}^{ab}{}_{J'}$ approach the asymptotes: $w^{ab}_{J'K'} = 1$ and $w^{ab}_{J'k} = 0$ for $k \neq K'$. In general, intervals R^a_J and/or R^b_K expand to $R^a_J \oplus \boldsymbol{a}$ and $R^b_K \oplus \boldsymbol{b}$; or remain as before; or new ART$_a$ and/or ART$_b$ categories are established.

Each graph in Fig. 11 shows, for each test set point a, the interval R^b_K predicted by the ART$_a$ category J selected by a. Recall that the maximum length of each interval R^b_k is 0.1, which constitutes the asymptotic approximation criterion, or confidence interval, once the training set has grown to a sufficiently large size.

Table II lists the number of ART$_a$ and ART$_b$ categories established in the simulations graphed in Fig. 11. Test set performance rate was determined by randomly selecting points $a \in [0, 1]$. Each point chose some ART$_a$ category J, which in turn predicted an ART$_b$ category K. If the length of the interval $R^b_K \oplus (f(a))$ was less than $(1 - \rho_b)$, that point was said to have met the matching criterion, or degree of approximation accuracy, determined by ρ_b. By Table II, the proportion of test set points that met this criterion grew from

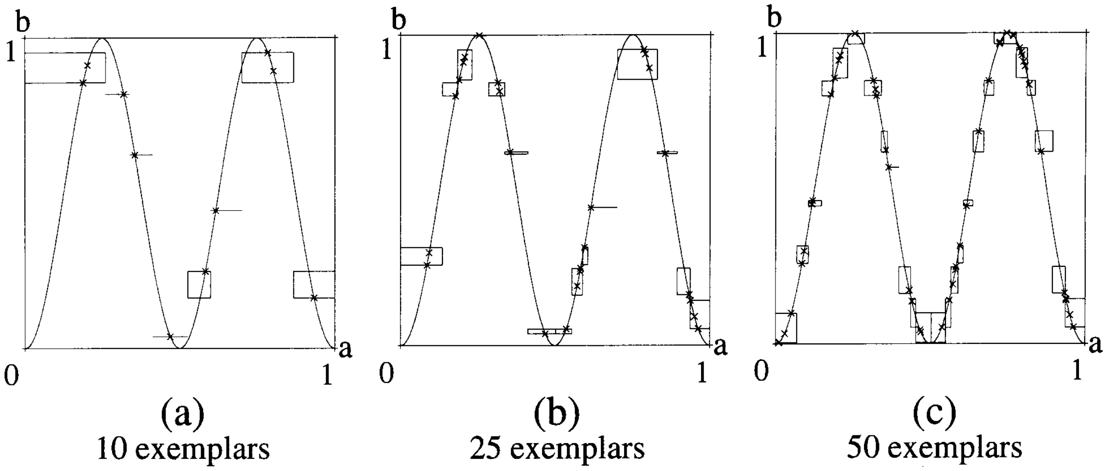


Fig. 11. Incremental approximation of a sinusoidal function as the number of training set exemplars increased from (a) 10 to (b) 25 to (c) 50. The simulation used fuzzy ARTMAP with complement coding, the conservative limit ($\alpha \cong 0$), fast learning ($\beta = 1$), and vigilance parameters $\overline{\rho}_a = 0, 0 < \rho_{ab} < 1$, and $\rho_b = 0.9$. Training set points (a) and test set ART$_b$ confidence intervals are shown on each graph. The maximum length of each confidence interval is $(1 - \rho_b) = 0.1$. See Table II.

TABLE II
INCREMENTAL FUNCTION APPROXIMATION

No. Training Exemplars	No. ART$_a$ Categories	No. ART$_b$ Categories	% Test Set Meeting Matching Criterion
10	8	7	31.9%
25	15	10	47.8%
50	26	11	72.3%

*ART$_b$ vigilance ($\rho_b = 0.9$) sets the matching criterion. After training on 50 exemplars, 72.3% of points $a \in [0, 1]$ choose an ART$_b$ node K such that $|R_K^b \oplus \boldsymbol{b}| \leq 1 - \rho_b = 0.1$, where $\boldsymbol{b} = (f(a))$. See Fig. 11.

31.9%, after training on ten exemplar pairs $(a, b) = (a, f(a))$, to 72.3% after training on 50 exemplar pairs.

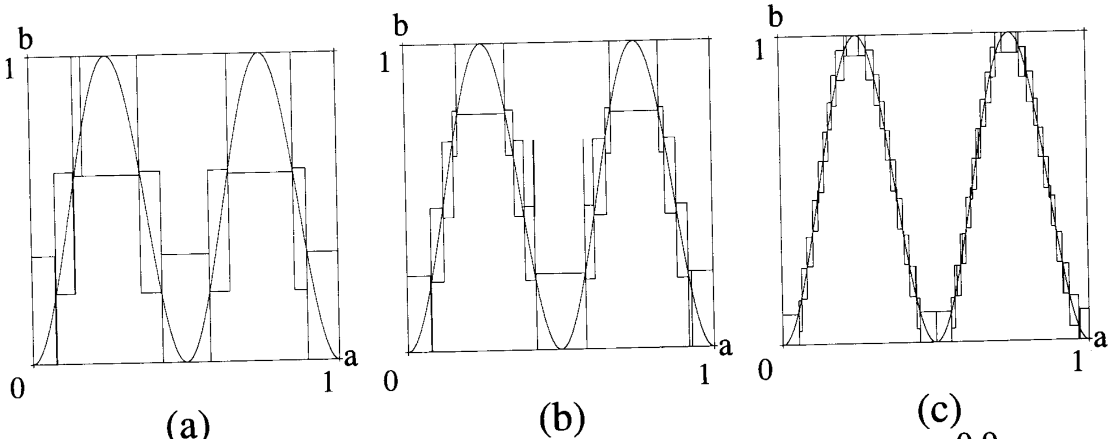
Fig. 12 shows the confidence intervals R_k^b of three function approximations, each with 1000 training set exemplars. In Fig. 11(a), ART$_b$ vigilance $\rho_b = 0.6$, which implies that the maximum length of each confidence interval is 0.4. By Table III, ten ART$_a$ nodes and three ART$_b$ nodes were used to establish this rough approximation. For 100% of the points a in the test set, $f(a)$ met the matching criterion. Thus learning had nearly reached its asymptote. If training were to continue indefinitely, some intervals R_j^a and R_k^b might expand slightly, but it is unlikely that new category nodes would be needed.

Parts (b) and (c) of Fig. 12 show how the function approximation improves as ρ_b is increased to 0.75 and to 0.9, respectively. With $\rho_b = 0.9$, the maximum length of each confidence interval R_k^b is 0.1. The stricter ART$_b$ matching criterion leads this system to establish more ART$_a$ and ART$_b$ categories than for the coarser approximations. In addition, 1.3% of test set points do not yet meet the matching criterion, which implies that a few more categories would be established before the system approached its asymptotic performance. Note that, in both (a) through (c) of Fig. 11 and (c) of Fig. 12, $\rho_b = 0.9$. Improvement from a 72.3% test set rate after 50 inputs (Fig. 11(c)) to a 98.7% rate after 1000 inputs (Fig. 12(c)) required 34 additional ART$_a$ categories but only two additional ART$_b$ categories.

IX. Simulation: Letter Image Recognition

Frey and Slate recently developed a benchmark machine learning task that they describe as a "difficult categorization problem" [17, 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 20 "fonts represent five different stroke styles (simplex, duplex, complex, and Gothic) and six different letter styles (block, script, italic, English, Italian, and German)" [17, 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. Thus the task is to learn to map from $\Re^{16}$ to $\Re^{26}$. The resulting letter image recognition file is archived in the UCI Repository of Machine Learning Databases and Domain Theories, maintained by D. Aha and P. 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 [18]–[20]. The training set consisted of 16 000 exemplars, with the remaining 4000 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 five 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 4000-item test set. The best performance (80.8%) was obtained using an integer input representation, a reward



(a)
$\rho_b = 0.6$

(b)
$\rho_b = 0.75$

(c)
$\rho_b = 0.9$

Fig. 12. Incremental approximation of a sinusoidal function for ART$_b$ vigilance parameters, with ρ_b equal to (a) 0.6, (b) 0.75, and (c) 0.9. In each simulation the fuzzy ARTMAP system was trained on 1000 randomly chosen points $a \in [0,1]$. Each graph shows the test set confidence intervals R_K^b selected by the test set points. The maximum lengths of these intervals are (a) 0.4, (b) 0.25, and (c) 0.1. Graph (c), with $\rho_b = 0.9$, is close to the asymptotic state of the three graphs in Fig. 11. See Table III.

TABLE III
Function Approximations, Each with 1000 Training Exemplars

ART$_b$ Vigilance ρ_b	No. ART$_a$ Categories	No. ART$_b$ Categories	% Test Set Meeting Matching Criterion
0.60	10	3	100.0%
0.75	17	5	99.9%
0.90	60	13	98.7%

*Stricter ART$_b$ matching criteria (larger ρ_b) cause the system to create more ART$_a$ and ART$_b$ categories. (See Fig.12.)

TABLE IV
Fuzzy ARTMAP Simulations of the Frey-Slate [17] Character Recognition Task

	% Correct Test Set Predictions	No. ART$_a$ Categories	No. Epochs
(a) $\alpha = 0.001$			
8 simulations			
Average	89.6%	799	6.00
Range	89.1% -90.8%	731-827	5-9
(b) $\alpha = 0.1$			
4 simulations			
Average	90.1%	825	6.25
Range	89.1%-91.1%	803-849	5-8
(c) $\alpha = 1.0$			
16 simulations			
Average	94.0%	1,023	5
Range	93.5-94.7%	988-1070	5

* The choice parameter α is equal to (a) 0.001, (b) 0.1, and (c) 1.0. Prediction rates are given for the 4000 item test set, after training on 16 000 exemplars.

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, which had 80.8% performance rate, ended with 1302 rules and eight 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 $|w_j^a|$ 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, which 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. Of the 28 Fuzzy ARTMAP simulations summarized in Table IV, the one with the best performance had a 94.7% correct prediction rate on the 4000-item test set, after five training epochs that created 1029 ART$_a$ categories. Thus the error rate (5.3%) was less than one-third that of the best simulation in the Frey–Slate comparative study (19.2%), or the lowest rates (18.4% and 17.3%) obtained with their alternative schemes. Moreover fuzzy ARTMAP simulations each created fewer than 1070 ART$_a$ categories, compared with the 1040–1302 final rules of the three genetic classifiers with the best performance rates. With voting, fuzzy ARTMAP reduced the error rate to 4.0% (Table V). Both the fuzzy ARTMAP simulation and the three genetic algorithm simulations with minimal error rates used five training epochs, as did the Fuzzy ARTMAP simulations in Table IV (c) and Table V (b, d, f). 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 V).

Table IV summarizes test set prediction rates on 28 fuzzy ARTMAP simulations, along with the number of ART$_a$ categories created and the number of epochs needed to reach asymptotic training set performance. Each simulation had a different, randomly chosen presentation order for the 16 000

TABLE V
VOTING STRATEGY APPLIED TO SETS OF THREE AND FIVE FUZZY ARTMAP SIMULATIONS OF THE FREY–SLATE CHARACTER RECOGNITION TASK

	% 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%	1,101	5
Range	93.4%–94.6%	990–1,070	5
Voting	96.0%		

* The choice parameter $\alpha = 0.1$ (a, b) or $\alpha = 1.0$ (c–f), and with training on one epoch (a, c, e) and five epochs (b, d, f). (a) Voting eliminated 30% of the individual simulation test set errors, which dropped from a three-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%

training set exemplars. The choice parameter α in (2) was set near the conservative limit at value $\alpha = 0.001$ in Table IV(a), and at the higher values $\alpha = 0.1$ in Table IV(b) and $\alpha = 1.0$ in Table IV(c). For $\alpha = 0.001$ and $\alpha = 0.1$, inputs were repeatedly presented in a given order until 100% training set performance was reached, which required from five to nine epochs. In order to make an exact comparison with the Frey–Slate simulations, training was stopped after five epochs for $\alpha = 1.0$ in Table IV(c), leaving training set performance below 100%. Both performance and the number of ART$_a$ categories increased with α. All simulations used fast learning, which creates a distinct ART$_a$ category structure for each input ordering. The number of ART$_a$ categories ranged from 731 to 1070 across the 28 simulations. All simulations used baseline vigilance $\overline{\rho_a} = 0$, which tends to minimize the number of ART$_a$ categories compared with higher values of $\overline{\rho_a}$. The remaining two vigilance parameters (ρ_{ab} and ρ_b) could be set to any number between 0 and 1, owing to the categorical nature of the ART$_b$ input vector and fast learning. In addition, preliminary studies had revealed that several of the 16 input features could be eliminated altogether without degrading performance. Simulations were thus carried out using a reduced ART$_a$ input vector that represented only the last 11 of the 16 features in the original input set, resulting in faster computation and decreased memory requirements.

Variations on the 28 simulations of Table IV gave similar performance results. For example in ten other simulations with $\alpha = 0.001$, the 16 000 inputs were trained for exactly five epochs instead of training each to 100% accuracy as in Table IV. On the average, the system used 811 ART$_a$ nodes and had a 90.6% test set prediction rate on these ten simulations. The test set performance rate was also similar when the system was trained on 19 000 items and tested on 1000 items. Values of α greater than 1.0 gave slightly improved performance but required more ART$_a$ nodes and more computation. Finally, fuzzy ARTMAP also performed well with on-line incremental training. With inputs selected at random, an early error rate of about 38% was achieved with 1250 training inputs. Each time the cumulative number of inputs doubled, the error rate was cut about in half, reaching 0.3% by input number 160 000. After 200 000 inputs (equivalent to about ten epochs), asymptotic performance was reached, with 100% correct prediction on each input.

Table V shows how voting consistently improves performance. In each group, with $\alpha = 0.1$ or $\alpha = 1.0$ and with one or five training epochs, fuzzy ARTMAP was run for three or five 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 V(a), for example, voting caused the error rate to drop to 8.8%, from a three-simulation average of 12.5%. With one training epoch, three-simulation voting eliminated about 30–35% of the test set errors (Table V(a) and V(c)), and five-simulation voting eliminated about 43% of the test set errors (Table V(e)). In the five-epoch simulations, where individual training set performance was close to 100%, three-simulation voting still reduced the test set error rate by about 25% (Table V(b) and V(d)) and five-simulation voting reduced the error rate by about 34% (Table V(f)). The best overall results were obtained with $\alpha = 1.0$ and five-epoch training, where voting reduced the five-simulation average error rate of 6.1% to a voting error rate of 4.0% (Table V(f)).

A final comparison between fuzzy ARTMAP and the genetic algorithms was made between the size $|\boldsymbol{w}_j^a|$ of the fuzzy ARTMAP weight vectors and the mean rule specificity, or number of non–wild card attributes, calculated by Frey and Slate. In fuzzy ARTMAP, a larger weight $\boldsymbol{w}_j^a$, which corresponds to a smaller hyperrectangle R_j^a, is less likely to be selected by a new item than a smaller weight, all other things being equal. Similarly, Frey and Slate estimate "that, in general, more specific rules will participate in the action [for category selection] less often than more general rules" [17, p. 171]. The Frey–Slate algorithm with a test-set prediction rate of 80.8% had a mean rule specificity of 8.02. This index was close to the average size (10.0) of the fuzzy ARTMAP weight vectors $\boldsymbol{w}_j^a$ in one simulation, with $\alpha = 1.0$, that used 1042 ART$_a$ nodes and had a 94.3% prediction rate. In this simulation, the number of ART$_a$ nodes predicting a given letter ranged from 19 nodes predicting L to 60 nodes predicting H, with an average of 39 nodes predicting a given letter. There was a strong correlation between the number of ART$_a$ nodes that, if chosen, predicted a letter and the error rate of that set of nodes. For example, the set of nodes predicting the letter H had an error rate of 11.2% and the set of nodes predicting the letter L had an error rate of 2.3%. Evidently, new categories are created as a difficult letter makes predictive errors during training.

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 one to five epochs, individual fuzzy ARTMAP systems had a robust prediction rate of 90% to 94% on the 4000-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%.

Acknowledgment

The authors wish to thank C. E. Bradford and D. J. Meyers for the valuable assistance in preparing the manuscript.

References

[1] G. A. Carpenter, S. Grossberg, and J. H. Reynolds, "ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network," *Neural Networks*, vol. 4, pp. 565–588, 1991. Tech. Rep. CAS/CNS-TR-91-001. Boston, MA: Boston University. Reprinted in [3].

[2] G. A. Carpenter, S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine," *Computer Vision, Graphics, and Image Processing*, vol. 37, pp. 54–115, 1987. Reprinted in [3].

[3] G. A. Carpenter and S. Grossberg, *Pattern Recognition by Self-Organizing Neural Networks*. Cambridge, MA: MIT Press, 1991.

[4] G. A. Carpenter, S. Grossberg, D. B. Rosen, "Fuzzy ART: An adaptive resonance algorithm for rapid, stable classification of analog patterns," in *Proc. Int. Joint Conf. Neural Networks*, vol. II, 411–420.

[5] G. A. Carpenter, S. Grossberg, and D. B. Rosen "Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system," *Neural Networks*, vol. 4, pp. 759–771, 1991. Tech. Rep. CAS/CNS-TR-91-015. Boston, MA: Boston University.

[6] B. Kosko, "Fuzzy entropy and conditioning," *Inform. Sci.*, vol. 40, pp. 165–174, 1986.

[7] L. Zadeh, "Fuzzy sets," *Inform. Contr.*, vol. 8, pp. 338–353, 1965.

[8] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum Press, 1981.

[9] S. Pal, and D. K. Dutta Majumder, *Fuzzy Mathematical Approach to Pattern Recognition*. New York: Wiley, 1986.

[10] G. A. Carpenter, S. Grossberg, and D. B. Rosen, "A neural network realization of fuzzy ART," *Tech. Rep.* CAS/CNS-TR-91-021. Boston, MA: Boston University, 1991.

[11] S. L. Salzberg, "Learning with nested generalized exemplars," Ph.D. (Technical Report TR-14-89), Department of Computer Science, Harvard University, Cambridge, MA, 1989.

[12] S. L. Salzberg, *Learning with Nested Generalized Exemplars*. Hingham, MA: Kluwer Academic, 1990.

[13] S. L. Salzberg, "A nearest hyperrectangle learning method," *Machine Learning*, vol. 6, pp. 251–276, 1991.

[14] P. Simpson, "Fuzzy min-max classification with neural networks," *Heuristics: J. Knowledge Eng.*, vol. 4, pp. 1–9, 1991.

[15] G. Wilensky, "Analysis of neural network issues: Scaling, enhanced nodal processing, comparison with standard classification," DARPA Neural Network Program Review, Oct. 29–30, 1990.

[16] K. J. Lang, and M. J. Witbrock, "Learning to tell two spirals apart," in *Proc. 1988 Connectionist Models Summer School*, 1989, pp. 52–59.

[17] P. W. Frey, and D. J. Slate, "Letter recognition using Holland-style adaptive classifiers," *Machine Learning*, vol. 6, pp. 161–182, 1991.

[18] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press, 1975.

[19] J. H. Holland, "Adaptive algorithms for discovering and using general patterns in growing knowledge bases," *Int. J. Policy Analysis and Information Systems*, vol. 4, pp. 217–240, 1980.

[20] J. H. Holland, "Escaping brittleness: The possibilities of general purpose machine learning algorithms applied to parallel rule-based systems," in *Machine Learning: An Artificial Intelligence Approach*, R. S. Michalski, J. G. Carbonell and T. M. Mitchell, Eds., vol. II. San Mateo, CA: Morgan Kaufmann, 1986.

[21] B. Moore, "ART 1 and pattern clustering," in *Proc. 1988 Connectionist Models Summer School*, 1989, pp. 174–185.

[22] D. E. Rumelhart, G. Hinton, and R. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing*. D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA: MIT Press, 1986.

[23] S. E. Fahlman, "Faster-learning variations on back-propagation: An empirical study," in *Proc. 1988 Connectionist Models Summer School*, 1989, pp. 38–51.

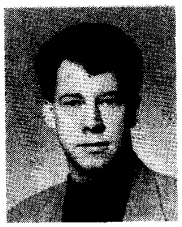
Gail A. Carpenter received the Ph.D. in mathematics in 1974. Since then she has carried out research on neural models of vision, nerve impulse generation (Hodgkin–Huxley equations), and complex biological rhythms. Her recent work has focused on the development of the adaptive resonance theory architectures (ART 1, 2, and 3) for self-organizing category learning and pattern recognition and on systems that incorporate ART modules into neural network architectures for incremental supervised learning (ARTMAP).

She is Professor of Cognitive and Neural Systems (CNS) and Mathematics at Boston University, and is the CNS director of graduate studies. Dr. Carpenter has served as vice president of the International Neural Network Society (1989), organization chair of the 1988 INNS annual meeting, and a member of the editorial boards of *Neural Networks, Neural Computation,* and *Brain Research.*

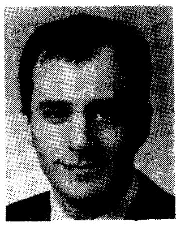
Stephen Grossberg is Wang Professor of Cognitive and Neural Systems at Boston University, where he is the founder and Director of the Center for Adaptive Systems, as well as the founder and Chairman of the Department of Cognitive and Neural Systems. He also founded and was the first president of the International Neural Network Society, and founded and served as coeditor in chief of the Society's journal, *Neural Networks.* His research includes content-addressable memories; associative learning; biological vision and image processing; adaptive pattern recognition; auditory and speech perception, adaptive sensory motor control; adaptive robotics; conditioning and attention; and biological rhythms.

Dr. Grossberg received his graduate training at Stanford University and Rockefeller University, and was a Professor at M.I.T. before assuming his present position at Boston University.

Natalya Markuzon is a second-year graduate student in the Department of Cognitive and Neural Systems at Boston University. Her current research interests are in the field of pattern recognition. She received the BA and MA degrees in mathematics from the Institute of Oil and Gas in Moscow, Russia.



John H. Reynolds is a Ph.D. student in the Department of Cognitive and Neural Systems at Boston University. Before joining the CNS program he received a bachelor's degree in economics from the University of Pennsylvania, after which he worked for an investment bank. Working with Prof. Carpenter and Prof. Grossberg at Boston University, he helped develop the ARTMAP architecture. His interests include reinforcement learning, limbic–cortical interactions in visual memory and attentional modulation, content addressable memories, and adaptive pattern recognition.



David B. Rosen (S'91) is currently completing work for the Ph.D. degree in the Department of Cognitive and Neural Systems at Boston University. His research interests include pattern recognition, neural networks, machine learning, statistical estimation and detection, function approximation, associative and content-addressable memories, and decision theory. He received a bachelor's degree in physics from Oberlin College, and an M.S. degree in physics in 1989 from the University of Minnesota, where he did research on the detection of proton decay. As part of his Ph.D. thesis research, he developed fuzzy ART and ART 2-A, with Prof. Carpenter and Prof. Grossberg.

Mr. Rosen is a member of the International Neural Network Society, the Classification Society of North America, and the International Student Society for Neural Networks.