1993-07

A Neural Network Architecture for Autonomous Learning, Recognition, and Prediction in a Nonstationary World

A NEURAL NETWORK ARCHITECTURE FOR AUTONOMOUS LEARNING, RECOGNITION, AND PREDICTION IN A NONSTATIONARY WORLD

Gail A. Carpenter and Stephen Grossberg

July, 1993

Technical Report CAS/CNS–93–049

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

A NEURAL NETWORK ARCHITECTURE FOR AUTONOMOUS LEARNING, RECOGNITION, AND PREDICTION IN A NONSTATIONARY WORLD

Gail A. Carpenter† and Stephen Grossberg‡

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

July, 1993

Technical Report CAS/CNS-TR-93-049
Boston, MA: Boston University

To appear in
An Introduction to Neural and Electronic Networks, Second Edition
Steven F. Zornetzer, Joel L. Davis, and Clifford Lau, Editors
New York: Academic Press, 1993

---

† Supported in part by ARPA (N00014-92-J-4015), British Petroleum (BP 89A-1204), 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 F49620-92-J-0225), ARPA (ONR N00014-92-J-4015), and the Office of Naval Research (ONR N00014-91-J-4100).

Acknowledgements: The authors wish to thank Cynthia Bradford, Robin Locke, and Diana Meyers for their valuable assistance in the preparation of the manuscript.

In a constantly changing world, humans are adapted to alternate routinely between attending to familiar objects and testing hypotheses about novel ones. We can rapidly learn to recognize and name novel objects without unselectively disrupting our memories of familiar ones. We can notice fine details that differentiate nearly identical objects and generalize across broad classes of dissimilar objects. This chapter describes a class of self-organizing neural network architectures—called ARTMAP—that are capable of fast, yet stable, on-line recognition learning, hypothesis testing, and naming in response to an arbitrary stream of input patterns (Carpenter, Grossberg, Markuzon, Reynolds, and Rosen, 1992; Carpenter, Grossberg, and Reynolds, 1991). The intrinsic stability of ARTMAP allows the system to learn incrementally for an unlimited period of time. System stability properties can be traced to the structure of its learned memories, which encode clusters of attended features into its recognition categories, rather than slow averages of category inputs. The level of detail in the learned attentional focus is determined moment-by-moment, depending on predictive success: an error due to over-generalization automatically focuses attention on additional input details, enough of which are learned in a new recognition category so that the predictive error will not be repeated.

An ARTMAP system creates an evolving map between a variable number of learned categories that compress one feature space (e.g., visual features) to learned categories of another feature space (e.g., auditory features). Input vectors can be either binary or analog. Computational properties of the networks enable them to perform significantly better in benchmark studies than alternative machine learning, genetic algorithm, or neural network models. Some of the critical problems that challenge and constrain any such autonomous learning system will next be illustrated. Design principles that work together to solve these problems are then outlined. These principles are realized in the ARTMAP architecture, which is specified as an algorithm. Finally, ARTMAP dynamics are illustrated by means of a series of benchmark simulations.

Critical Problems to be Solved by an Autonomous Learning System

ARTMAP performance success is based on a set of design principles that are derived from an analysis of learning by an autonomous agent in a nonstationary environment (Table 1). Realization of these principles enables a self-organizing ARTMAP system to learn, categorize, and make predictions about a changing world, as follows.

Rare Events: A successful autonomous agent must be able to learn about rare events that have important consequences, even if these rare events are similar to a surrounding cloud of frequent events that have different consequences. *Fast learning* is needed to pick up a rare event on the fly. For example, a rare medical condition could be either a unique case or the harbinger of a new epidemic. A slightly different chemical assay could either be a routine variation or predict the biological activity of a new drug. Many feedforward neural network systems, such as back propagation, require a form of slow learning that tends to average over similar event occurrences. ARTMAP can rapidly group or single out events, depending on their predictive outcomes.

Large Nonstationary Databases: Rare events typically occur in a nonstationary environment whose event statistics may change rapidly and unexpectedly through time. Individual events may also occur with variable probabilities and durations, and arbitrarily large numbers of events may need to be processed. Each of these factors tends to destabilize

Table 1: AUTONOMOUS LEARNING AND CONTROL IN A NONSTATIONARY WORLD

An ARTMAP system can reconcile conflicting requirements and autonomously learn about:

RARE EVENTS
- requires FAST learning

LARGE NONSTATIONARY DATABASES
- requires STABLE learning

MORPHOLOGICALLY VARIABLE EVENTS
- requires MULTIPLE SCALES of generalization (fine/coarse)

MANY-TO-ONE AND ONE-TO-MANY RELATIONSHIPS
- requires categorization and naming for expert knowledge

To realize these properties, ARTMAP systems:

PAY ATTENTION
- ignore masses of irrelevant data

TEST HYPOTHESES
- discover predictive constraints hidden in data streams

CHOOSE BEST ANSWERS
- quickly select globally optimal solution at any stage of learning

CALIBRATE CONFIDENCE
- measure on-line how well a hypothesis matches the data

DISCOVER RULES
- identify transparent if-then relations at each learning stage

SCALE
- preserve all desirable propertie in arbitrarily large problems

the learning process within feedforward algorithms. New learning in such systems tends to unselectively wash away the memory traces of old, but still useful, knowledge. Using such an algorithm, for example, learning new faces could erase the memory of a parent's face. More generally, learning a new type of expertise could erase the memory of previous expert knowledge. ARTMAP contains a *self-stabilizing memory* that permits accumulating knowledge to be stored reliably in response to arbitrarily many events in a nonstationary environment under incremental learning conditions. Learning may continue until the system's full memory capacity, which can be chosen arbitrarily large, is exhausted.

Morphologically Variable Types of Events: In many environments, some information, including rule-like inferences, is coarsely defined whereas other information is precisely characterized. Otherwise expressed, the morphological variability of the data may change through time. For example, we may recognize one photograph as an animal, while seeing a similar one as a picture of our own pet. Under autonomous learning conditions, a system typically has to constantly adjust how coarse the generalization, or compression, of particular types of data should be. Multiple scales of generalization, from fine to coarse, need to be available on an as-needed basis. ARTMAP automatically adjusts its scale of generalization to match the morphological variability of the data, based on predictive success. The network embodies a Minimax Learning Rule that conjointly minimizes predictive error and maximizes generalization using only the information that is locally available under incremental learning conditions in a nonstationary environment. This property has been used to suggest how the inferotemporal cortex can learn to recognize both fine and coarse information about the world (Carpenter and Grossberg, 1993), as demonstrated by neurophysiological experiments of Desimone (1992), Harries and Perrett (1991), Miller, Li, and Desimone (1991), Mishkin (1982), and Spitzer, Desimone, and Moran (1988), among others.

Many-to-One and One-to-Many Relationships: In ARTMAP learning, many-to-one code compression occurs in two stages, categorization and naming. For example, during categorization of printed letter fonts, many similar exemplars of the same printed letter may establish a single recognition category, or compressed representation (Figure 1). Different printed letter fonts or written exemplars of the letter may establish additional categories. Each of these categories carries out a many-to-one map of exemplar into category. During naming, all of the categories that represent the same letter may be associatively mapped into the letter name, or prediction. Compressed many-to-one maps are hereby constructed from both unsupervised (categorization) and supervised (naming) learning.

Conversely, one-to-many learning is also used to build up expert knowledge about an object or event. A single visual image of a particular animal may, for example, lead to learning that predicts: animal, dog, beagle, and my dog Rover (Figure 2). A computerized record of a patient's medical check-up may lead to a series of predictions about the patient's health.

In feedforward networks, the attempt to learn more than one prediction about a single input leads to unselective forgetting of previously learned predictions, for the same reason that these algorithms become unstable in response to nonstationary data. In particular, error-based learning systems, including multi-layer perceptrons such as back propagation (Rosenblatt, 1958; Rumelhart, Hinton, and Williams, 1986; Werbos, 1974), find it difficult, if not impossible, to solve the critical problems described above.

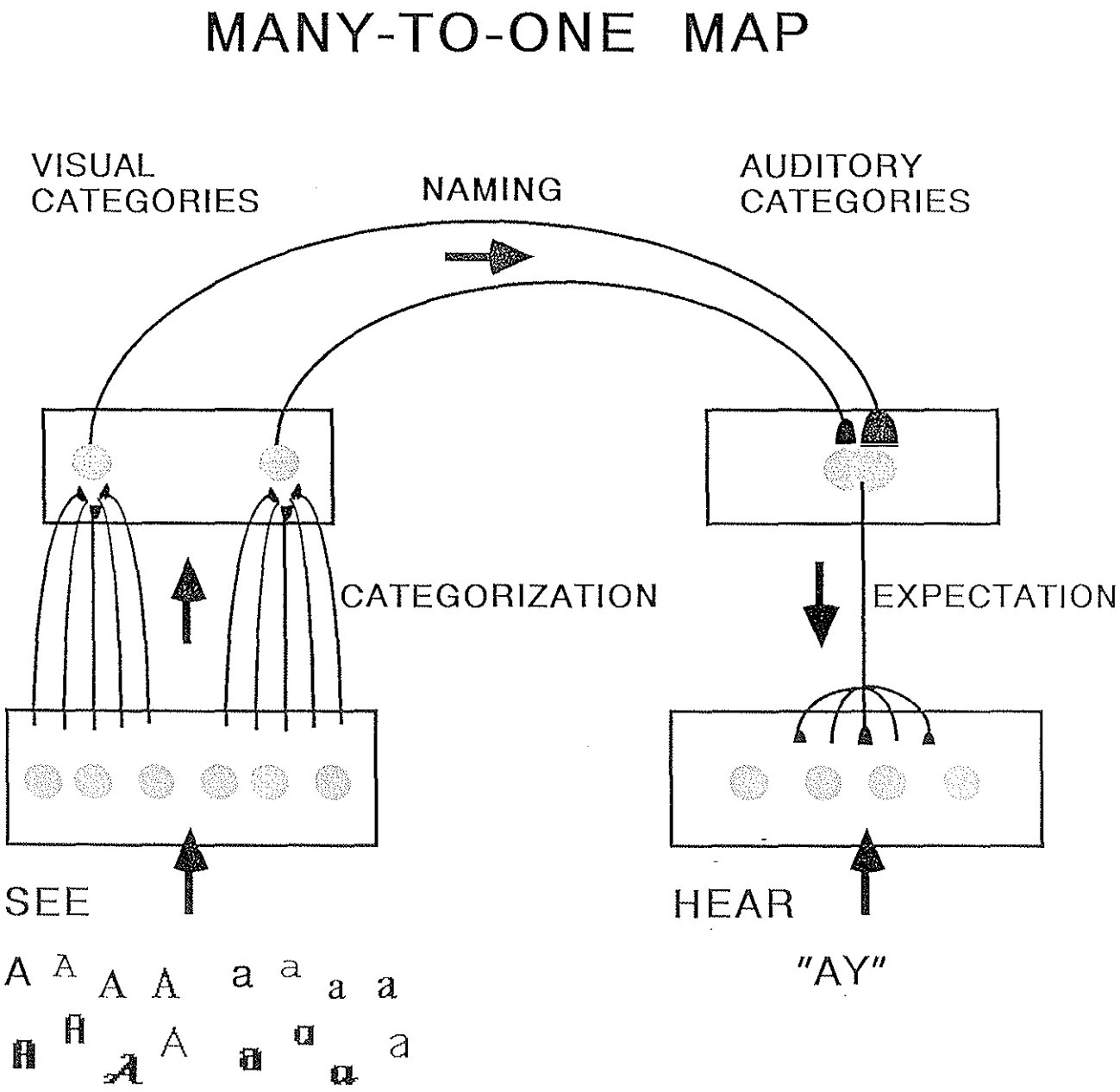


Figure 1. Many-to-one learning combines categorization of many exemplars into one category, and labelling of many categories with the same name.

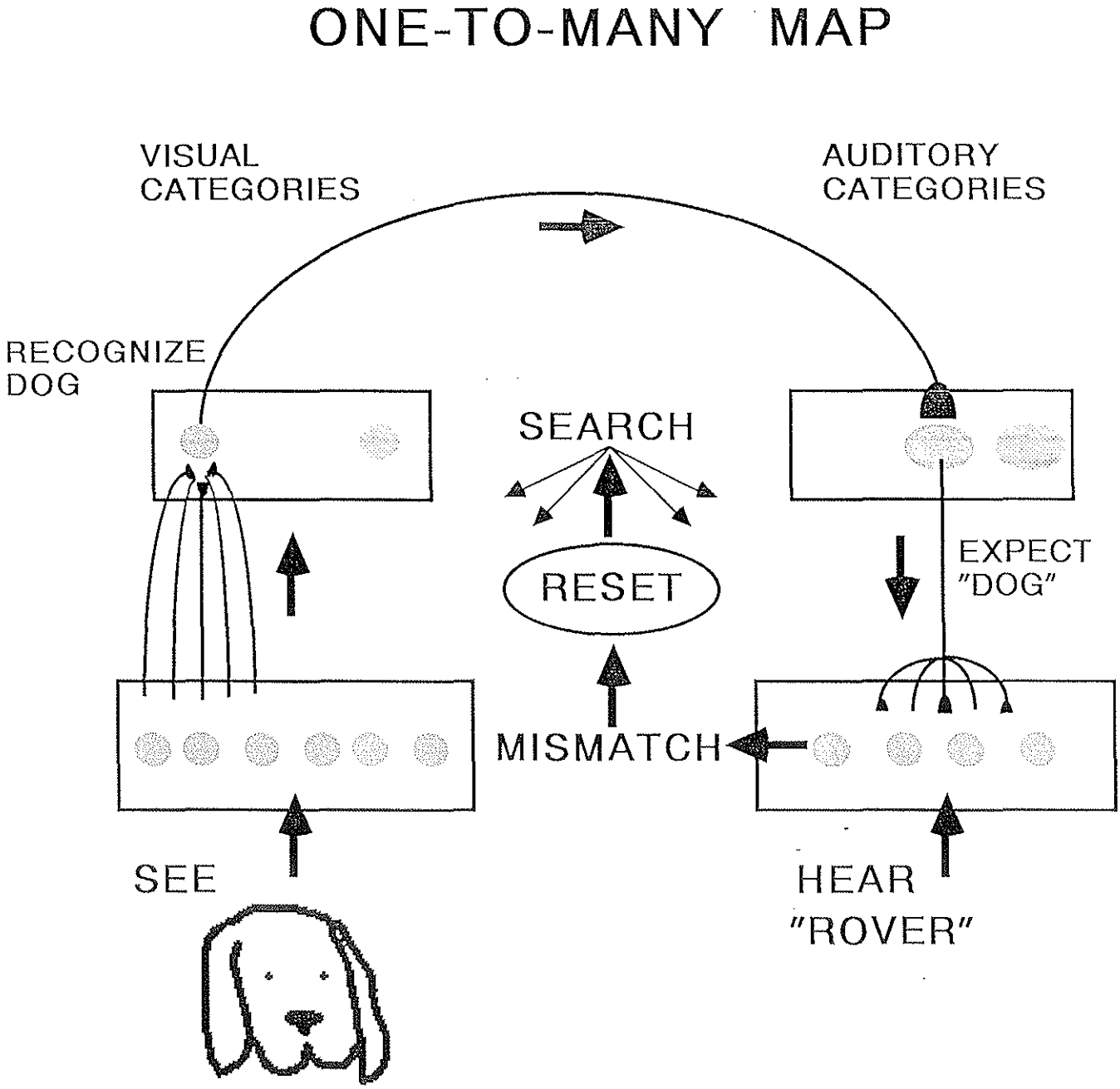


Figure 2. One-to-many learning enables one input vector to be associated with many output vectors. If the system predicts an output that is disconfirmed at any given stage of learning, the predictive error drives a memory search for a new category to associate with the new prediction without degrading its previous knowledge about the input vector.

ARTMAP Design Principles

ARTMAP systems solve the critical design problems because they implement a qualitatively different set of heuristics than error-based learning systems, as follows (Table 1).

Pay Attention: ARTMAP learns top-down expectations (also called prototypes, primes, or queries) that allow the system to ignore masses of irrelevant distributed data. These queries "test the hypothesis" that is embodied by a recognition category, or symbol, as they suppress features not in the prototypical attentional focus. When one object is recognized as "dog" versus "Rover," distinct top-down expectations focus attention on distinct feature clusters. ARTMAP hereby embodies properties of intentionality. A large mismatch between a bottom-up input vector and a top-down expectation (Figure 2) can drive an adaptive memory search that carries out hypothesis testing for a better category, as described below.

Hypothesis Testing and Match-Based Learning: ARTMAP actively searches for recognition categories, or hypotheses, whose top-down expectations provide an acceptable match to bottom-up data. The top-down expectation learns a prototype that focuses attention upon that cluster of input features that it deems relevant. If no available category, or hypothesis, provides a good enough match, then selection and learning of a new category and top-down expectation is automatically initiated. When the search discovers a category that provides an acceptable match, the system locks into an attentive resonance through which the distributed input and its symbolic category are bound together. During this resonantly bound state, the input exemplar refines the adaptive weights of the category based on any new information in the attentional focus. Thus ARTMAP carries out match-based learning, rather than error-based learning, because a category modifies its previous learning only if its top-down expectation matches the input vector well enough to risk changing its defining characteristics. Otherwise, hypothesis testing selects a new category on which to base learning of a novel event, thereby preserving information in both the new and the old categories.

Choose Globally Best Symbolic Answer: In many learning algorithms, as learning proceeds, local minima, or less than optimal solutions, are selected. In ARTMAP, at any stage of learning, an input exemplar first selects the category whose top-down expectation provides the globally best match. This top-down expectation hereby acts as a prototype for the class of all the input exemplars that its category represents. After learning self-stabilizes, every input directly selects the globally best matching category without any search. This category symbolically represents all the inputs that share the same prototype. Before learning self-stabilizes, familiar events gain direct access to the globally best category without any search, even if they are interspersed with unfamiliar events that drive hypothesis testing for better matching categories. A lesion in the *orienting subsystem*, that mediates the hypothesis testing, or memory search, process, leads to a memory disorder that strikingly resembles clinical properties of medial temporal amnesia in humans and monkeys after lesions of the hippocampal formation (Carpenter and Grossberg, 1993). These and related data properties provide support for the hypothesis that the hippocampal formation carries out an orienting subsystem function as one of its several functional roles.

Learn Prototypes and Exemplars: The learned prototype represents the cluster of input features that the category deems relevant based upon its past experience. The prototype represents the features to which the category "pays attention". In cognitive psychology, an input pattern is called an exemplar. A fundamental issue in cognitive psychology con-

cerns whether the brain learns prototypes or exemplars. Some argue that the brain learns prototypes, or abstract types of knowledge, such as being able to recognize that a particular object is a face or an animal. Others have argued that the brain learns individual exemplars, or concrete types of knowledge, such as being able to recognize a particular face or a particular animal. Recently it has been increasingly realized that some sort of hybrid system is needed that can acquire both types of knowledge (Smith, 1990). ARTMAP is such a hybrid system. It uses the Minimax Learning Rule to control how abstract or concrete—how fuzzy—a category becomes in order to conjointly minimize predictive error and maximize predictive generalization.

Calibrate Confidence: A confidence measure, called *vigilance*, calibrates how well an exemplar must match the prototype that it selects. Otherwise expressed, vigilance measures how well the chosen hypothesis must match the data. If vigilance is low, even poorly matching exemplars can then be incorporated into one category, so compression and generalization by that category are high. The symbol here is more abstract. If vigilance is high, then even good matches may be rejected, and hypothesis testing may be initiated to select a new category. In this case, few exemplars activate the same category, so compression and generalization are low. In the limit of very high vigilance, prototype learning reduces to exemplar learning.

The Minimax Learning Rule is realized by adjusting the vigilance parameter in response to a predictive error. Vigilance is first low, to maximize compression. When a predictive error occurs, vigilance is increased just enough to initiate hypothesis testing to discover a better category, or hypothesis, with which to match the data. In this way, a minimum amount of generalization is sacrificed to correct the error. This process is called *match tracking* because vigilance tracks the degree of match between exemplar and prototype in response to a predictive error.

IF-THEN Rule Discovery: At any stage of learning, a user can translate the learned weights of an ARTMAP system into a set of IF-THEN rules that completely characterize the decisions of the system. These rules evolve as ARTMAP is exposed to new inputs. Suppose, for example, that n visual categories are associated with the auditory prediction "AY." Backtrack from prediction "AY" along the associative pathways whose adaptive weights have learned to connect the n visual categories to this prediction (Figure 1). Each of these categories codes a "reason" for predicting "AY." The prototype of each category embodies the set of features, or constraints, whose binding together constitutes that category's "reason." The IF-THEN rule takes the form: IF some of the features of any of these n categories are found bound together, within the fuzzy constraints that would lead to selection of that category, THEN the prediction "AY" holds. Keeping in mind that ARTMAPs carry out hypothesis testing and memory search to discover these rules, we can see that ARTMAPs are a type of self-organizing production system (Laird, Newell, and Rosenbloom, 1987) that evolves adaptively from individual input-output experiences, as in case-based reasoning.

IF-THEN rules of ARTMAP can be extracted from the system at any stage of the learning process. This property is particularly important in applications such as medical diagnosis from a large database of patient records, where doctors may want to study the rules by which the system reaches its diagnostic decisions. Some of these rules may already be familiar to the doctors. Others may represent novel constraint combinations (symptoms, tests, treatments, ...) which the doctors could then evaluate for their possible medical significance. This property also sheds light on how humans believe that brains somehow realize rule-like

behavior although brain anatomy is not algorithmically structured in a traditional sense. The Minimax Learning Rule determines how abstract these rules will become in response to any prescribed environment. Typical databases generate a mixture of a few broad rules, with few constraints and many exemplars, plus a set of more highly specified special cases (Carpenter, Grossberg, and Reynolds, 1991).

Table 2 summarizes some medical and other benchmark studies that compare the performance of ARTMAP with alternative recognition and prediction models. Three of these benchmarks are summarized below. These and other benchmarks are described elsewhere in greater detail (Carpenter, Grossberg, and Iizuka, 1992; Carpenter, Grossberg, Markuzon, Reynolds, and Rosen, 1992; Carpenter, Grossberg, and Reynolds, 1991).

Properties Scale: One of the most serious deficiencies of many algorithms is that their desirable properties tend to break down as small toy problems are generalized to large-scale problems. In contrast, all of the desirable properties of ARTMAP scale to arbitrarily large problems. Recall, however, that ARTMAP solves a particular class of problems, not all problems of learning or intelligence. The categorization and inference problems that ARTMAP does handle well are, however, core problems in many intelligent systems, and include technology bottlenecks for many alternative approaches.

ARTMAP Architecture

Each ARTMAP system includes a pair of ART modules (ART_a and ART_b), as in Figure 3. 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. As noted above, this is accomplished by realizing a Minimax Learning Rule that 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 (Figure 4).

Parameter ρ_a calibrates the minimum confidence that ART_a must have in a recognition category, or hypothesis, that is 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 the minimal confidence ρ_a by the least amount needed to trigger hypothesis testing at ART_a, using a mechanism called *match tracking*. Match tracking sacrifices the minimum amount of generalization necessary to correct the predictive error. Match tracking increases the criterion confidence just enough to trigger hypothesis testing. 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.

An ARTMAP simulation algorithm will now be summarized. When input components are binary, the ARTMAP system is constructed from ART 1 component modules (Carpenter

Table 2: BENCHMARK STUDIES

Database benchmark:

MACHINE LEARNING (90-95 % correct)

ARTMAP (100% correct)

Training set an order of magnitude smaller.

Medical database:

STATISTICAL METHOD (60% correct)

ARTMAP (91% correct)

Incrementally improvement.

Transparant "rules" from critical feature clusters.

Letter recognition database:

GENETIC ALGORITHM (82% correct)

ARTMAP (96% correct)

Database benchmarks:

BACKPROPAGATION (10,000 - 20,000 training epochs)

ARTMAP (1-5 epochs)

Used in applications where other algorithms fail

e.g., Boeing CAD Group Technology (T. Caudell et al.)

Part design reuse and inventory compression.

Need fast stable learning and search of a huge (16 million) and continually growing nonstationary parts inventory.

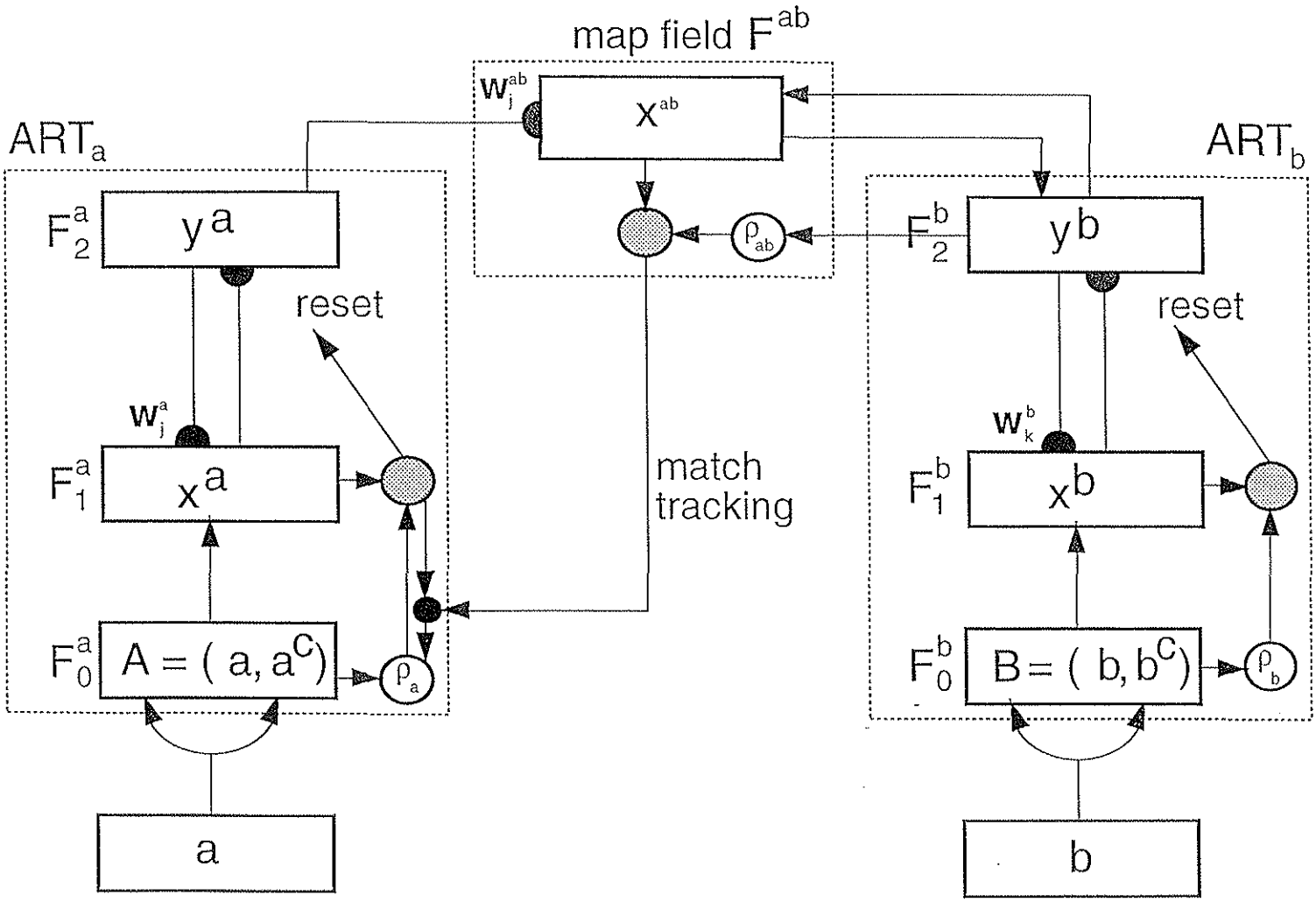


Figure 3. Fuzzy ARTMAP architecture. The ART_a complement coding preprocessor transforms the M_a-vector $\mathbf{a}$ into the $2M_a$-vector $\mathbf{A} = (\mathbf{a}, \mathbf{a}^c)$ at the ART_a field F_0^a. $\mathbf{A}$ is the input vector to the ART_a field F_1^a. Similarly, the input to F_1^b is the $2M_b$-vector $(\mathbf{b}, \mathbf{b}^c)$. When 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 $\mathbf{b}$ or to a previously uncommitted ART_a category node.

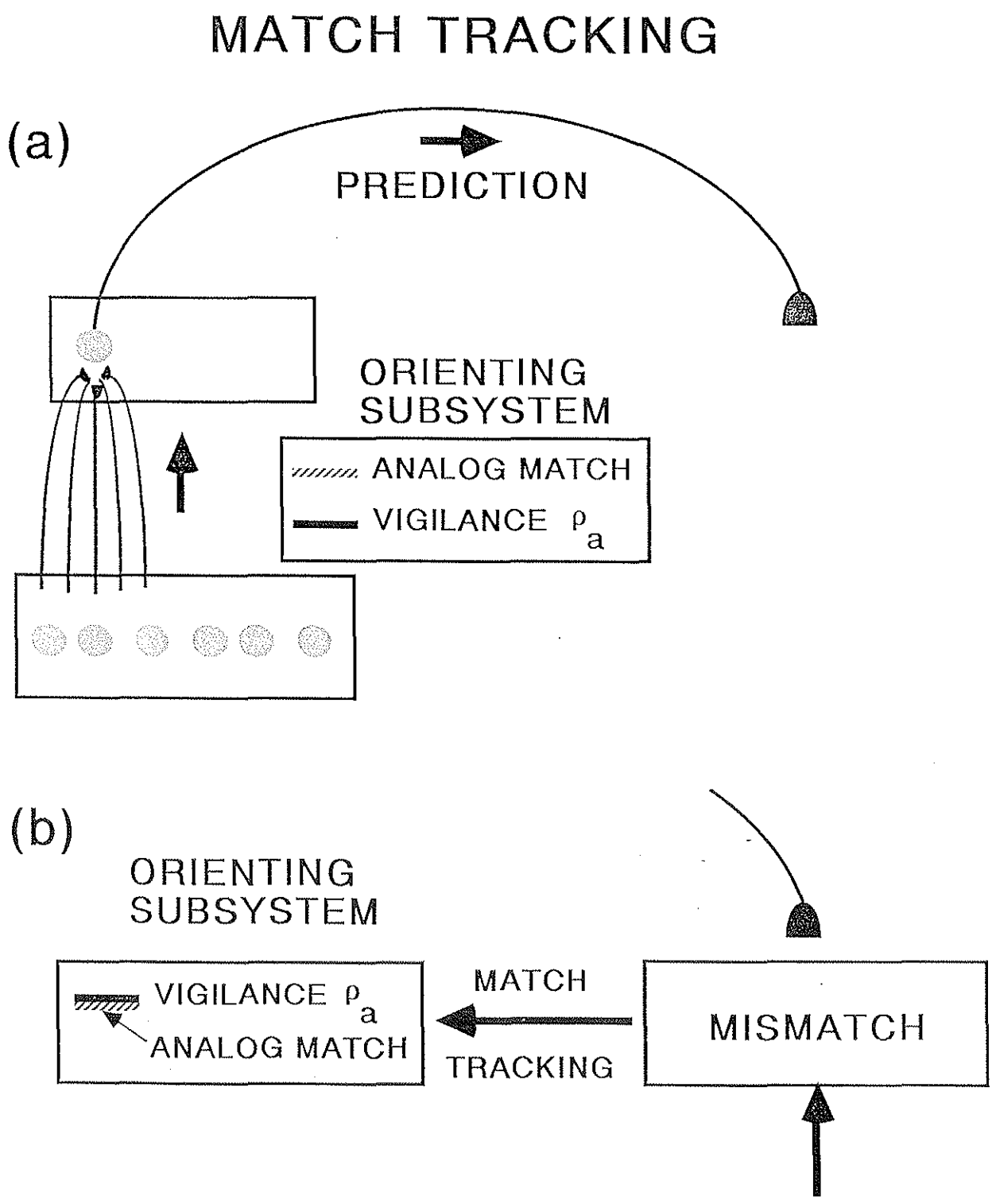


Figure 4. Match tracking: (a) A prediction is made by ART_a when the baseline vigilance ρ_a is less than the analog match value. (b) A predictive error at ART_b increases the baseline vigilance value of ART_a until it just exceeds the analog match value, and thereby triggers hypothesis testing that searches for a more predictive bundle of feature to which to attend.

and Grossberg, 1987). ART 1 dynamics are simulated via a series of operations that include binary intersection ($\bigcap$). When input components are analog (real-valued), the set-theoretic intersection operator can be replaced by a fuzzy intersection ($\wedge$), or componentwise minimum (Zadeh, 1965). Binary ART 1 then is thereby transformed into fuzzy ART (Carpenter, Grossberg, and Rosen, 1991), and binary ARTMAP (Carpenter, Grossberg, and Reynolds, 1991) becomes fuzzy ARTMAP (Carpenter, Grossberg, Markuzon, Reynolds, and Rosen, 1992). Algorithms for the more general systems, fuzzy ART and fuzzy ARTMAP, will now be specified.

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. 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 Long-Term Memory (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 the ART 1 neural network.

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 operator $\wedge$ is defined by

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

and where the city-block 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 $\mathbf{a}$. 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

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

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. The interactions mediated by the map field F^{ab} may be operationally characterized as follows.

$\mathbf{ART}_a$ and $\mathbf{ART}_b$

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 3). 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 $\mathbf{w}_j^a \equiv (w_{j1}^a, w_{j2}^a, \ldots, w_{j,2M_a}^a)$ 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), $\mathbf{x}^{ab} = \mathbf{y}^b \wedge \mathbf{w}_J^{ab} = \mathbf{0}$ 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}. A predictive mismatch is detected when:

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

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. After match tracking,

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

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 $F_2^a \rightarrow F^{ab}$ 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.

The Geometry of Fuzzy ART

Fuzzy ARTMAP dynamics will be illustrated below by a benchmark simulation problem, circle-in-the-square. The low dimensions of this problem ($M_a = 2, N_a = 1$) allow the evolving category structure to be illustrated graphically. To do this, a geometric interpretation of

fuzzy ART will now be outlined. For definiteness, let the input set consist of 2-dimensional vectors $\mathbf{a}$ preprocessed into the 4-dimensional complement coding form. Thus

$$\mathbf{I} = (\mathbf{a}, \mathbf{a}^c) = (a_1, a_2, 1 - a_1, 1 - a_2). \tag{22}$$

In this case, each category j has a geometric representation as a rectangle R_j, as follows. Following (22), the weight vector $\mathbf{w}_j$ can be written in complement coding form:

$$\mathbf{w}_j = (\mathbf{u}_j, \mathbf{v}_j^c), \tag{23}$$

where $\mathbf{u}_j$ and $\mathbf{v}_j$ are 2-dimensional vectors. Let vector $\mathbf{u}_j$ define one corner of a rectangle R_j and let $\mathbf{v}_j$ define another corner of R_j (Figure 5a). The size of R_j is defined to be

$$|R_j| \equiv |\mathbf{v}_j - \mathbf{u}_j|, \tag{24}$$

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

In a fast-learn fuzzy ART system, with $\beta = 1$ in (11), $\mathbf{w}_J^{(\text{new})} = \mathbf{I} = (\mathbf{a}, \mathbf{a}^c)$ when J is an uncommitted node. The corners of $R_J^{(\text{new})}$ are then given by $\mathbf{u}_J = \mathbf{a}$ and $\mathbf{v}_J = (\mathbf{a}^c)^c = \mathbf{a}$. Hence $R_J^{(\text{new})}$ is just the point $\mathbf{a}$. Learning increases the size of each R_j. In fact the size of R_j grows as the size of $\mathbf{w}_j$ shrinks during learning. The maximum size of R_j is limited by the size of the vigilance parameter, with $|R_j| \leq 2(1 - \rho)$. During each fast-learning trial, R_J expands to $R_J \oplus \mathbf{a}$, the minimum rectangle containing R_J and $\mathbf{a}$ (Figure 5b). The corners of $R_J \oplus \mathbf{a}$ are given by $\mathbf{a} \wedge \mathbf{u}_J$ and $\mathbf{a} \vee \mathbf{v}_J$, where the fuzzy AND (intersection) operator $\wedge$ is defined by (3); and the fuzzy OR (union) operator $\vee$ is defined by

$$(\mathbf{p} \vee \mathbf{q})_i \equiv \max(p_i, q_i) \tag{25}$$

(Zadeh, 1965). Hence, by (24), the size of $R_J \oplus \mathbf{a}$ is given by

$$|R_J \oplus \mathbf{a}| = |(\mathbf{a} \vee \mathbf{v}_J) - (\mathbf{a} \wedge \mathbf{u}_J)|. \tag{26}$$

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

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

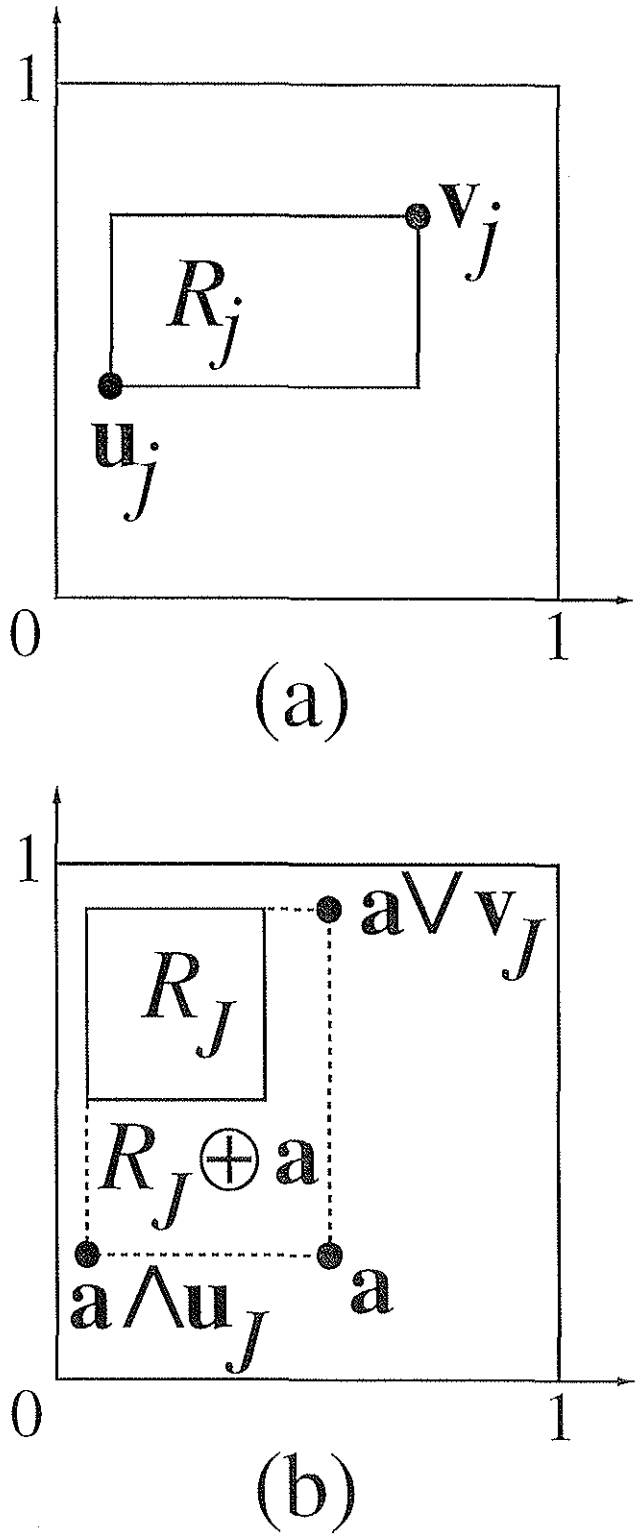


Figure 5. Fuzzy ART weight representation. (a) In complement coding form with $M = 2$, each weight vector $\mathbf{w}_j$ has a geometric interpretation as a rectangle R_j with corners $(\mathbf{u}_j, \mathbf{v}_j)$. (b) During fast learning, R_J expands to $R_J \oplus \mathbf{a}$, the smallest rectangle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

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 6 and 7. As training set size increased from 100 exemplars (Figure 6a) to 100,000 exemplars (Figure 6d) 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 $\mathbf{w}_j^{ab}$ 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. Figure 7 shows the ART_a category rectangles R_j^a established in each simulation of Figure 6. Initially, large R_j^a estimated large areas as belonging to one or the other category plus 3 point rectangles created near the decision boundary, to correct errors (Figure 7a). Additional R_j^a's improved accuracy, especially near the boundary of the circle (Figure 7d). The map can be made arbitrarily accurate provided the number of ART_a nodes is allowed to increase as needed. As in Figure 5 each rectangle R_j^a corresponds to the 4-dimensional weight vector $\mathbf{w}_j^a = (\mathbf{u}_j^a, (\mathbf{v}_j^a)^c)$, were $\mathbf{u}_j^a$ and $\mathbf{v}_j^a$ are plotted as the lower-left and upper-right corners of R_j^a, respectively.

Figure 8 depicts the response patterns of fuzzy ARTMAP on another series of circle-in-the-square simulations. The simulations used the same training sets as in Figure 6, but with each training set input presented for as many epochs as were needed to achieve 100% predictive accuracy on the training set. In each case, test set predictive accuracy increased, as did the number of ART_a category nodes. For example, with 10,000 exemplars, 1-epoch training used 50 ART_a nodes to give 96.7% test set accuracy (Figure 6c). The same training set, after 6 epochs, used 89 ART_a nodes to give 98.3% test set accuracy (Figure 8c).

Figure 6 showed a test set error rate that is reduced from 11.4% to 2.0% as training set size increases from 100 to 100,000 in 1-epoch simulations. Figure 8 showed 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. An ARTMAP *voting strategy* provides a third way to eliminate test set errors. The voting strategy assumes a fixed set of training exemplars, with the input ordering randomly assembled before each individual simulation. After the 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 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% (Figure 9c). In other words, test set errors were reduced from an average individual rate of 9.5% to a voting rate of 6.1%. Figure 9d indicates the number of votes cast for each test set point, and hence reflects variations in predictive confidence across different regions. Voting by more

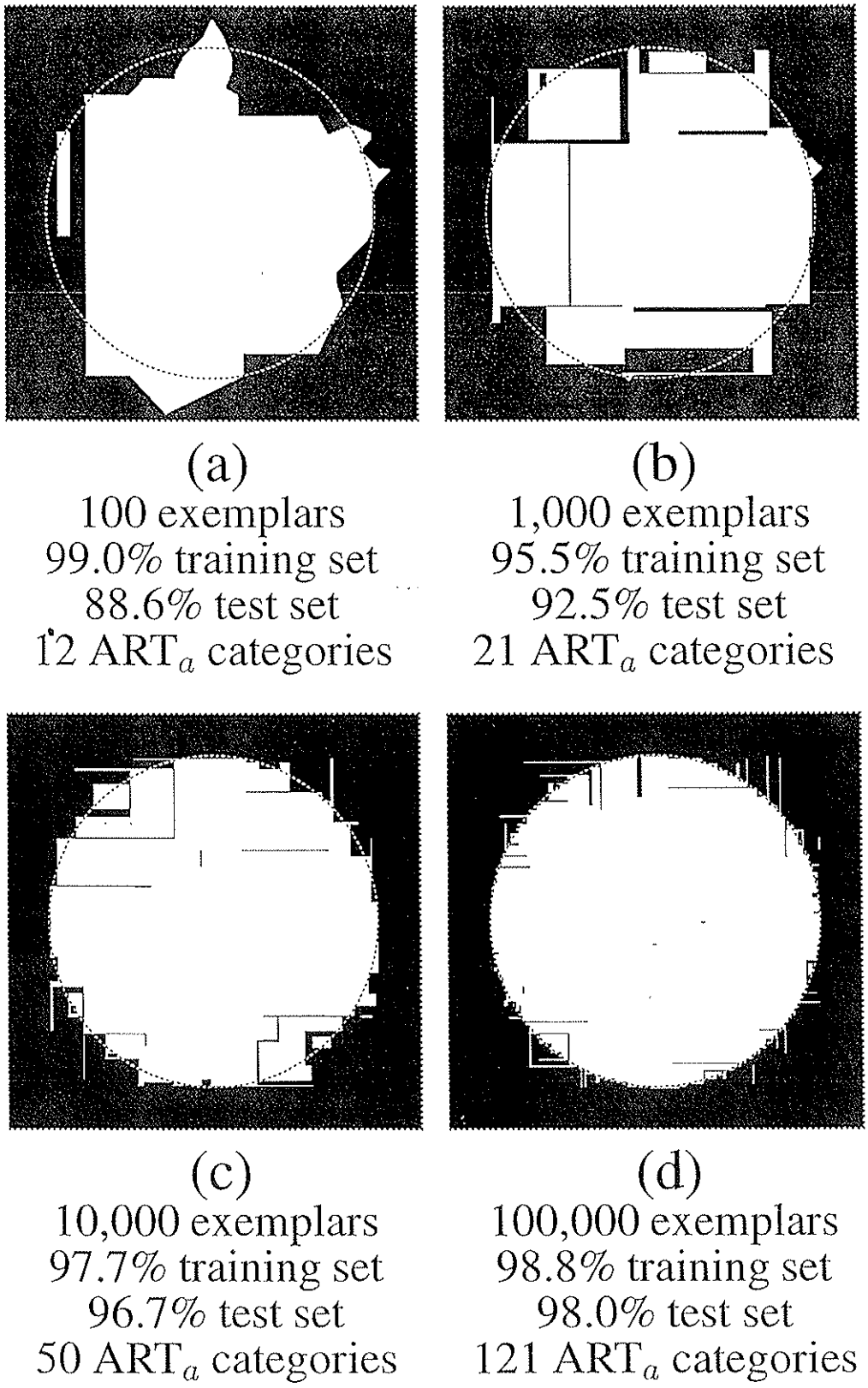
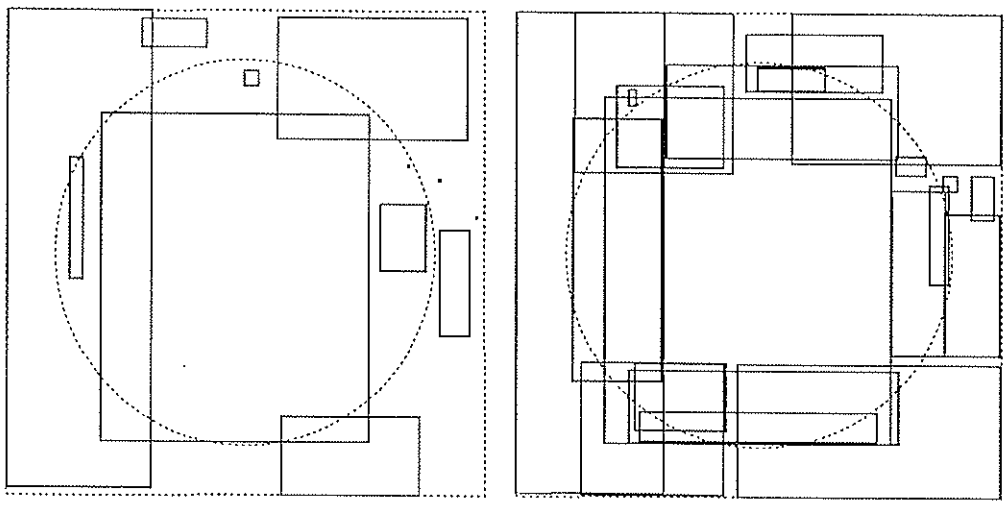
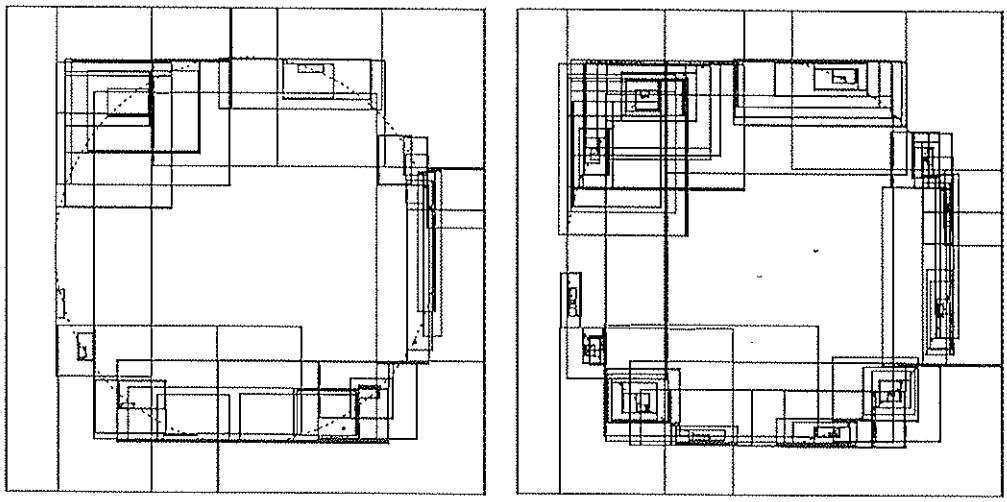


Figure 6. Circle-in-the-square test set response patterns after 1 epoch of fuzzy ARTMAP training on (a) 100, (b) 1,000, (c) 10,000, and (d) 100,000 randomly chosen training set points. Test set points in white areas are predicted to lie inside the circle and points in black areas are predicted to lie 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)
100 exemplars
99.0% training set
88.6% test set
12 ART$_a$ categories

(b)
1,000 exemplars
95.5% training set
92.5% test set
21 ART$_a$ categories



(c)
10,000 exemplars
97.7% training set
96.7% test set
50 ART$_a$ categories

(d)
100,000 exemplars
98.8% training set
98.0% test set
121 ART$_a$ categories

Figure 7. Fuzzy ARTMAP category rectangles R_j^a for the circle-in-the-square simulations of Figure 6. Small rectangles are created near the map discontinuities as the error rate drops toward 0.

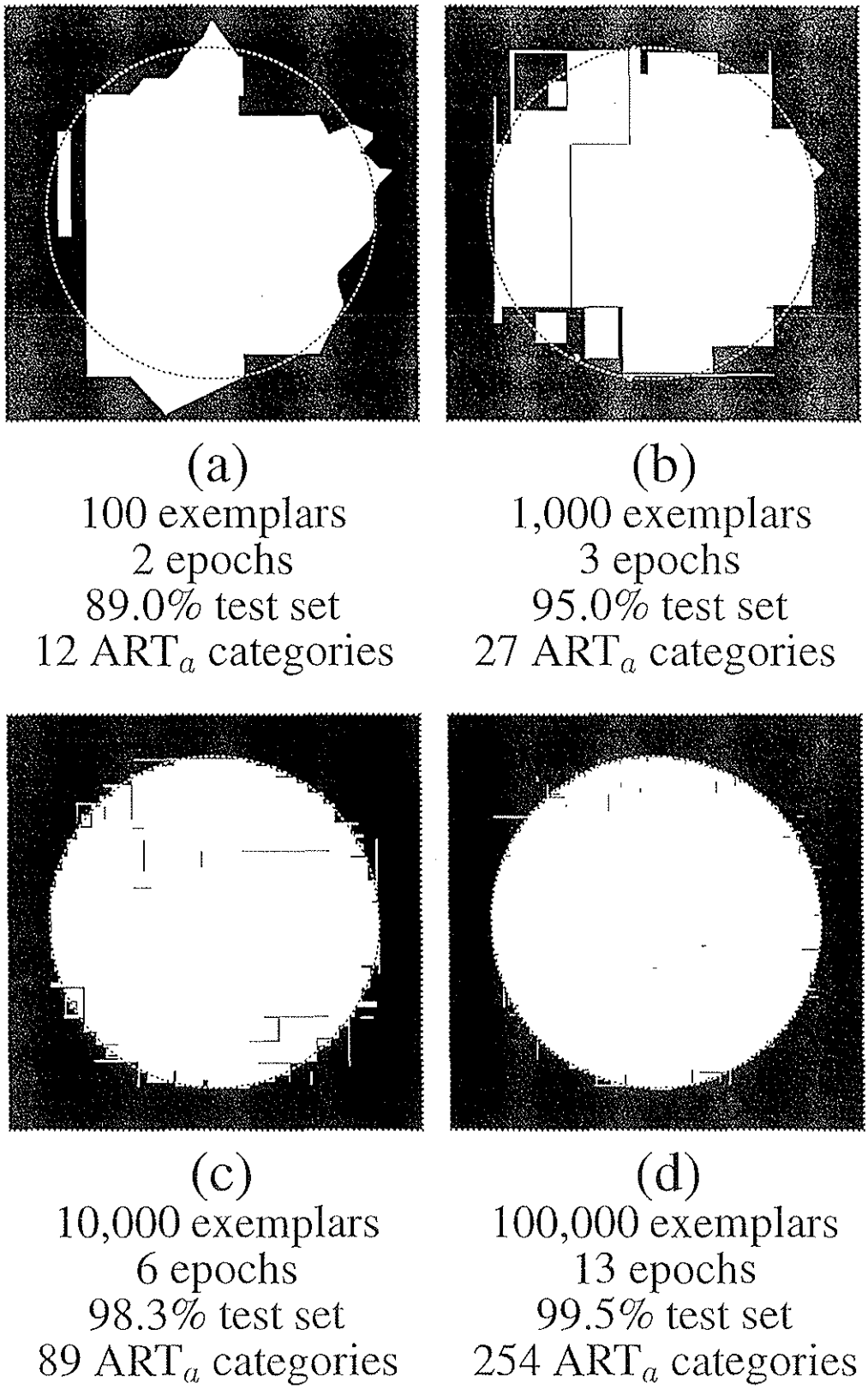


Figure 8. Circle-in-the-square test set response patterns with exemplars repeatedly presented until the system achieved 100% correct prediction on (a) 100, (b) 1,000, (c) 10,000, and (d) 100,000 training set points. Training sets were the same as those used for Figures 6 and 7. Training to 100% accuracy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

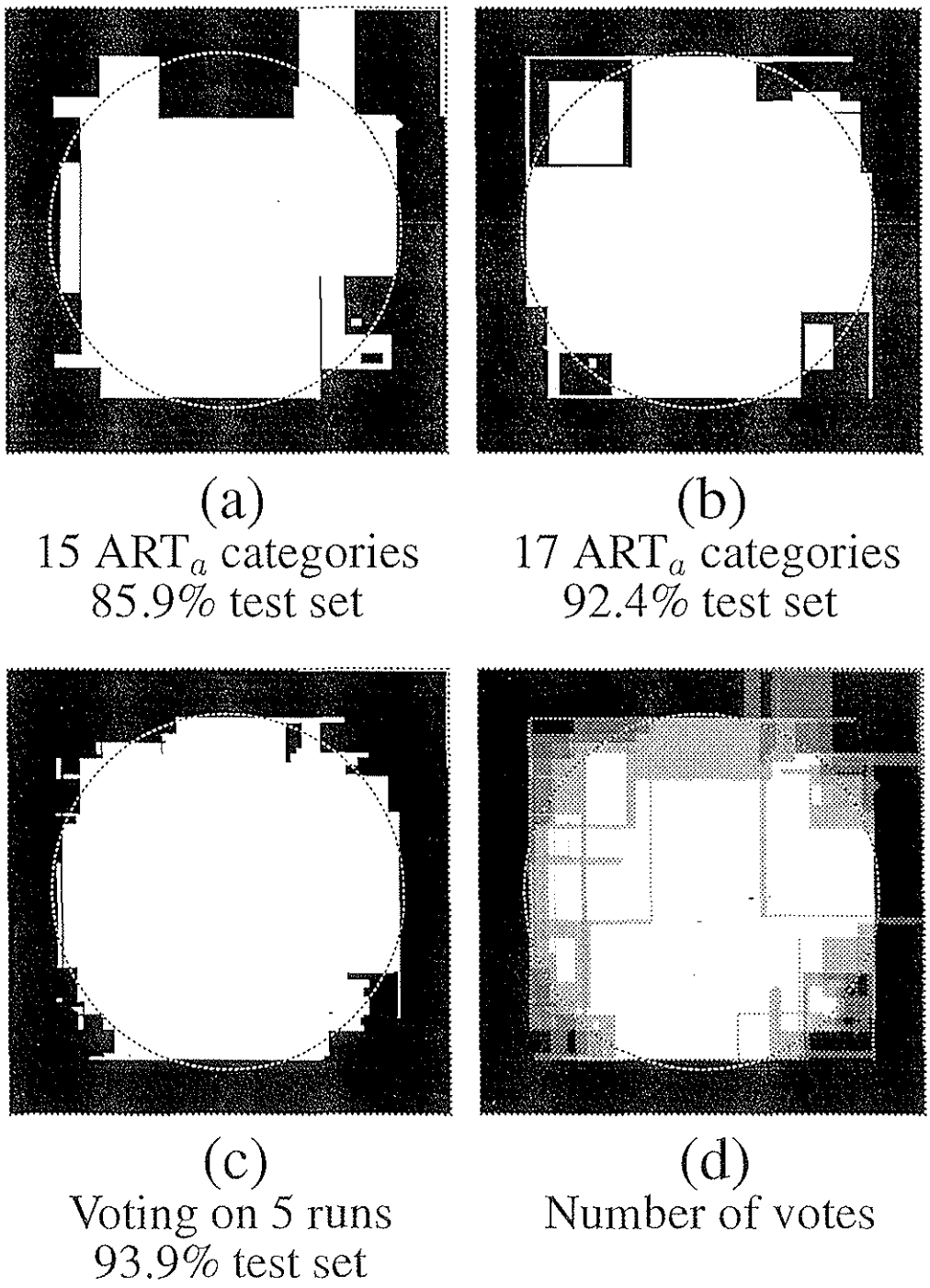


Figure 9. Circle-in-the-square response patterns for a fixed 1,000-item training set. (a) Test set responses after training on inputs presented in random order. After 1 epoch that used 15 ART$_a$ nodes, test set prediction rate was 85.9%, the worst of 5 runs. (b) Test set responses after training on inputs presented in a different random order. After 1 epoch that used 17 ART$_a$ nodes, test set prediction rate was 92.4%, 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.

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 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 Figure 6b, 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 6c).

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

$$\mathbf{A} = (a_1, a_2, 1 - a_1, 1 - a_2), \tag{27}$$

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

$$\mathbf{b} = \begin{cases} (1) & \text{if } \mathbf{a} \text{ is inside the circle} \\ (0) & \text{otherwise.} \end{cases} \tag{28}$$

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

$$\mathbf{B} = \begin{cases} (1,0) & \text{if } \mathbf{a} \text{ is inside the circle} \\ (0,1) & \text{otherwise.} \end{cases} \tag{29}$$

The fuzzy ARTMAP simulations used fast learning, defined by (11) with $\beta = 1$; the choice parameter $\alpha \cong 0$ (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.

Two Analog ARTMAP Benchmark Studies: Letter and Written Digit Recognition

As summarized in Table 2, fuzzy ARTMAP has been benchmarked against a variety of machine learning, neural network, and genetic algorithms with considerable success. An illustrative study used a benchmark machine learning task that Frey and Slate (1991) developed and described 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 (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 (Figure 10). 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 (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

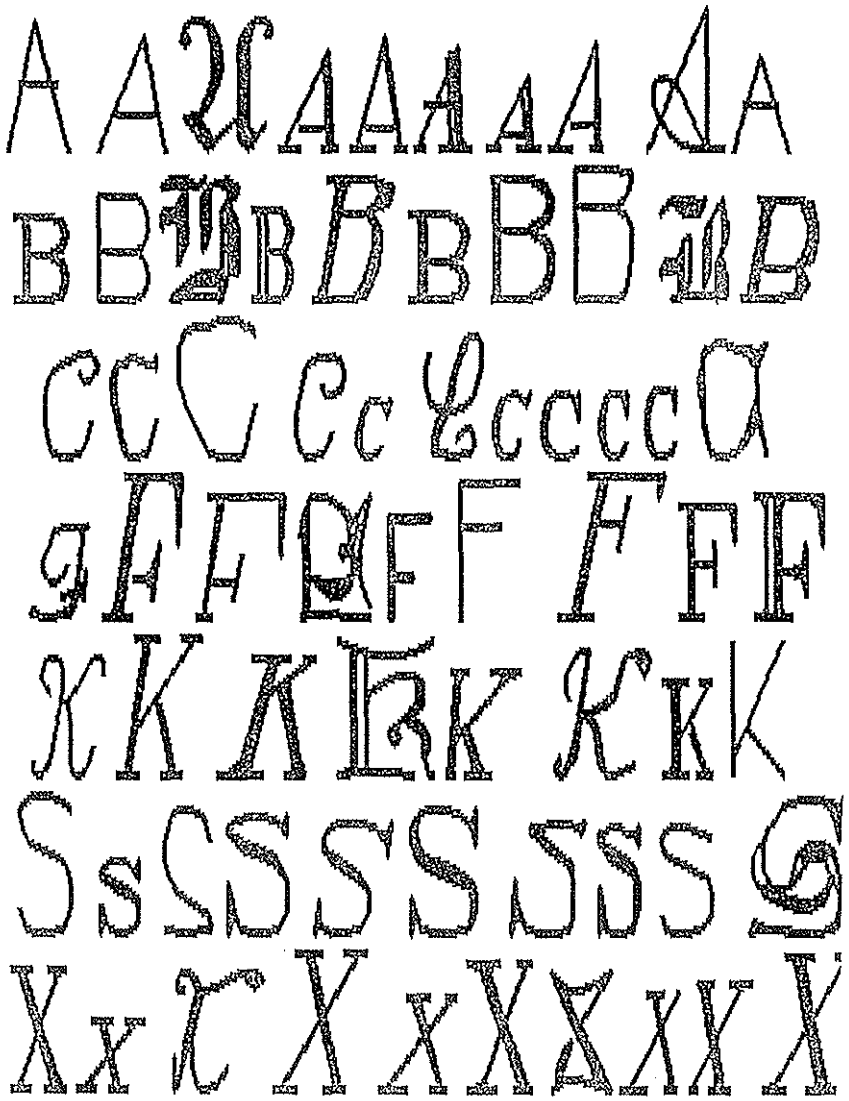


Figure 10. Illustrative letter fonts used by Frey and Slate (1991).

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 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. In particular, 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. The ARTMAP voting strategy consistently eliminated 25%–43% of the errors, giving a robust prediction rate of 92%–96%. 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. 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 five epochs.

Rapid learning was also found in a benchmark study of written digit recognition, where the correct prediction rate on the test set after one epoch reached over 99% of its best performance (Carpenter, Grossberg, and Iizuka, 1992). In this study, fuzzy ARTMAP was tested along with back propagation and a self-organizing feature map. Voting yielded fuzzy ARTMAP average performance rates on the test set of 97.4% after an average number of 4.6 training epochs. Back propagation achieved its best average performance rates of 96% after 100 training epochs. Self-organizing feature maps achieved a best level of 96.5%, again after many training epochs.

In summary, on a variety of benchmarks, fuzzy ARTMAP has demonstrated much faster learning and better performance compared to alternative machine learning, genetic, or neural network algorithms. In addition, fuzzy ARTMAP can be used in applications where many other adaptive pattern recognition algorithms cannot perform well. These are the classes of applications where very large nonstationary databases need to be rapidly organized into stable variable-compression categories under real-time autonomous learning conditions.

Concluding Remarks

Fuzzy ARTMAP is one of a rapidly growing family of attentive self-organizing learning hypothesis testing, and prediction systems that have evolved from the biological theory of cognitive information processing of which ART forms an important part (Carpenter and Grossberg, 1991). ART modules have found their way into such diverse applications as the

control of mobile robots, a Macintosh system that adapts to user behavior, diagnostic monitoring systems for nuclear plants, learning and search of airplane part inventories, medical diagnosis, 3-D visual object recognition, musical analysis, seismic recognition, sonar recognition, and laser radar recognition (Baloch and Waxman, 1991; Caudell, Smith, Johnson, Wunsch, and Escobedo, 1991; Gjerdingen, 1990; Goodman *et al.*, 1992; Kayvan, Durg, and Rabelo, 1993; Johnson, 1993; Seibert and Waxman, 1991). All of these applications exploit the ability of ART systems to rapidly learn to classify large databases in a stable fashion, to calibrate their confidence in a classification, and to focus attention upon those featural groupings that they deem to be important based upon their past experience. We anticipate that the growing family of supervised ARTMAP systems will find an even broader range of applications due to their ability to adapt the number, shape, and scale of their category boundaries to meet the on-line demands of large nonstationary databases.

REFERENCES

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

Carpenter, G.A. and Grossberg, S. (1987). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, **37**, 54–115. Reprinted in Carpenter, G.A. and Grossberg, S. (Eds.), **Pattern recognition by self-organizing neural networks**. Cambridge, MA: MIT Press, 1991.

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

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

Carpenter, G.A., Grossberg, S., and Iizuka, K. (1992). Comparative performance measures of fuzzy ARTMAP, learned vector quantization, and back propagation for handwritten character recognition. **Proceedings of the international joint conference on neural networks**, Baltimore, **I**, 794–799. Piscataway, NJ: IEEE Service Center.

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

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

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

Caudell, T., Smith, S., Johnson, C., Wunsch, D., and Escobedo, R. (1991). An industrial application of neural networks to reusable design. **Adaptive neural systems**, Technical Report BCS-CS-ACS-91-001, Seattle, WA: The Boeing Company, pp. 185–190.

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

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

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

Goodman, P.H., Kaburlasos, V.G., Egbert, D.D., Carpenter, G.A., Grossberg, S., Reynolds, J.H., Hammermeister, K., Marshall, G., and Grover, F. (1992). Fuzzy ARTMAP neural network prediction of heart surgery mortality. **Proceedings of the Wang Institute research conference: Neural networks for learning, recognition, and control**. Boston, MA: Boston University, p. 48.

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

Harries, M.H. and Perrett, D.I. (1991). Visual processing of faces in temporal cortex: Physiological evidence for a modular organization and possible anatomical correlates. *Journal of Cognitive Neuroscience*, **3**, 9–24.

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.

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

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

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

Miller, E.K., Li, L., and Desimone, R. (1991). A neural mechanism for working and recognition memory in inferior temporal cortex. *Science*, **254**, 1377-1379.

Mishkin, M. (1982). A memory system in the monkey. *Philosophical Transactions Royal Society of London* B, **298**, 85–95.

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

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

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.

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

Smith, E.E. (1990). In D.O. Osherson and E.E. Smith (Eds.), **An invitation to cognitive science**. Cambridge, MA: MIT Press.

Spitzer, H., Desimone, R., and Moran, J. (1988). Increased attention enhance both behavioral and neuronal performance. *Science*, **240**, 338–340.

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

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.

.