
Pergamon

Neural Networks

Neural Networks 11 (1998) 323–336

Contributed article

ARTMAP-IC and medical diagnosis: Instance counting and inconsistent cases

Gail A. Carpenter*, Natalya Markuzon

Center for Adaptive Systems and Department of Cognitive and Neural Systems, Boston University, 677 Beacon Street, Boston, MA 02215, USA

Received 24 June 1996; accepted 30 June 1997

Abstract

For complex database prediction problems such as medical diagnosis, the ARTMAP-IC neural network adds distributed prediction and category instance counting to the basic fuzzy ARTMAP system. For the ARTMAP match tracking algorithm, which controls search following a predictive error, a new version facilitates prediction with sparse or inconsistent data. Compared to the original match tracking algorithm (MT +), the new algorithm (MT −) better approximates the real-time network differential equations and further compresses memory without loss of performance. Simulations examine predictive accuracy on four medical databases: Pima Indian diabetes, breast cancer, heart disease, and gall bladder removal. ARTMAP-IC results are equal to or better than those of logistic regression, K nearest neighbour (KNN), the ADAP preceptron, multisurface pattern separation, CLASSIT, instance-based (IBL), and C4. ARTMAP dynamics are fast, stable, and scalable. A voting strategy improves prediction by training the system several times on different orderings of an input set. Voting, instance counting, and distributed representations combine to form confidence estimates for competing predictions.

Keywords: Automated medical prediction; Adaptive Resonance Theory; ARTMAP-IC; ARTMAP; Instance counting; Match tracking; Voting; Neural network

1. Neural networks and medical diagnosis

Neural networks, statistical methods, and machine learning algorithms are currently being tested on many medical prediction problems, with the goal of developing algorithms for accurate automatic diagnostic assistants. Generally, neural networks have performed at least as well as other methods, with coronary artery disease and breast cancer among the most widely studied databases. For example, in a well publicized study, Baxt (1991) used backpropagation to identify myocardial infarction; on a coronary artery disease database, Rosenberg et al. (1993) found performance of a radial basis function network to be comparable with that of human experts and superior to various backpropagation methods; and for breast cancer detection, researchers have successfully applied backpropagation (Floyd et al., 1994; Sahiner et al., 1995), ART 2 and fractal analysis (Downes, 1994), the neocognitron (Lo et al., 1995), convolution neural networks (Petrick et al., 1995), and decision trees (Bohren et al., 1995).

ARTMAP neural networks (Carpenter & Grossberg, 1991; Carpenter et al., 1991a, 1992) for supervised learning, recognition, and prediction have recently been used in a wide variety of applications. This paper introduces ARTMAP-IC, which adds to the basic ARTMAP system new capabilities designed to solve computational problems that frequently arise in medical database prediction. One such problem is inconsistent cases, where identical input vectors correspond to cases with different outcomes. ARTMAP-IC modifies the ARTMAP search algorithm to allow the network to encode inconsistent cases, and combines instance counting during training with distributed category representation during testing to obtain probabilistic predictions, even with fast learning and only one training epoch. Performance of ARTMAP-IC, named for instance counting and inconsistent cases, is tested on medical prediction problems by comparing results with those reported in four benchmark database studies. Methods compared include logistic

* Requests for reprints should be sent to G. A. Carpenter. E-mail: gail@cns.bu.edu.

0893–6080/98/$19.00
PII S0893-6080(97)00067-1

regression (Howell, 1992), the preceptron-like ADAP model (Smith, 1962), K nearest neighbor (KNN) (Duda & Hart, 1973), multisurface pattern separation (Mangasarian, 1968), the unsupervised CLASSIT algorithm (Gennari et al., 1989), the instance-based classifiers IB1, IB2, and IB3 (Aha et al., 1991), and the decision tree C4 (Quinlan, 1986). Medical records used in these studies are the Pima Indian diabetes data set (Smith et al., 1988), a University of Wisconsin breast cancer data set, a V.A. Hospital heart disease data set, and a Medicare cholecystectomy (gall bladder removal) data set.

Section 2 introduces the family of ARTMAP architectures, including fuzzy ARTMAP, ART-EMAP, and ARTMAP-IC. Section 3 analyzes the match tracking search process, comparing the new algorithm (MT−) with the original (MT+). Voting (Section 4), distributed prediction by a Q-max rule (Section 5), and instance counting (Section 6) augment computational capabilities of the basic ARTMAP network. Complete ARTMAP-IC implementation algorithms for training and testing (Section 7) characterize the network used in the simulations (Section 8) that compare performance of ARTMAP variations with benchmark results on four medical database problems.

2. ART and ARTMAP neural networks

ARTMAP networks for supervised learning self-organize mappings from input vectors, representing features such as patient history and test results, to output vectors, representing predictions such as the likelihood of an adverse outcome following an operation. The original binary ARTMAP (Carpenter et al., 1991a) incorporates two unsupervised ART 1 modules (Carpenter & Grossberg, 1987), ART_a and ART_b, that are linked by a *map field* F^{ab}. At the map field the network forms associations between categories via outstar learning and triggers search, via the ARTMAP match tracking rule, when a training set input fails to make a correct prediction. Match tracking increases the ART_a vigilance parameter ρ_a in response to predictive error at ART_b. Fuzzy ARTMAP (Carpenter et al., 1992) substitutes fuzzy ART (Carpenter et al., 1991b) for ART 1 (Fig. 1). ART-EMAP (Carpenter & Ross, 1993, 1995) uses distributed category representation to improve fuzzy ARTMAP performance. ARTMAP-IC extends this sequence with an instance counting procedure and a new match tracking algorithm that consistently improve both predictive accuracy and code compression, compared to the basic

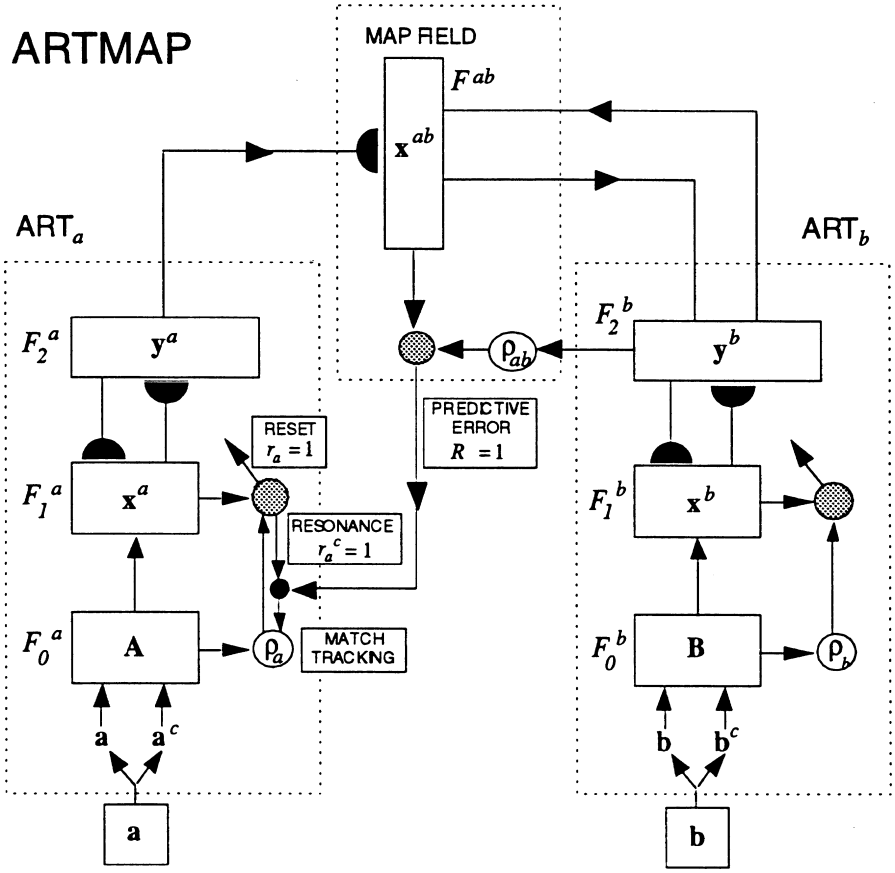

Fig. 1. ARTMAP architecture. The ART_a complement coding preprocessor transforms the M_a-vector $\mathbf{a}$ into the $2M_a$-vector $\mathbf{A} = (\mathbf{a}, \mathbf{a}^c)$ at the ART_a field F_0^a. $\mathbf{A}$ is the input vector to the ART_a field F_1^a. Similarly, the input to F_1^b is the $2M_b$-vector $\mathbf{B} = (\mathbf{b}, \mathbf{b}^c)$. When ART_b disconfirms a prediction of ART_a, map field inhibition induces the match tracking process. Match tracking raises the ART_a vigilance ρ_a to just above the F_1^a-to-F_0^a match ratio $|\mathbf{x}_a|/|\mathbf{A}|$. This triggers an ART_a search which leads either to an ART_a category that correctly predicts $\mathbf{b}$ or to a previously uncommitted ART_a category node (Carpenter et al., 1991).

ARTMAP and ART-EMAP networks. These added capabilities also allow ARTMAP-IC to encode predictions of inconsistent cases in the training set, giving good test set performance on various medical diagnosis problems.

Many applications of supervised learning systems such as ARTMAP are classification problems, where the trained system tries to predict a correct category given a test set input vector. A prediction might be a single category or distributed as a set of scores or probabilities. The ARTMAP-IC algorithm below (Section 7) outlines a procedure for applying ART learning and prediction to this problem, which does not require a full ART_b architecture (Fig. 2). In the algorithm an input $\mathbf{a} = (a_1 \ldots a_i \ldots a_M)$ learns to predict an outcome $\mathbf{b} = (b_1 \ldots b_k \ldots b_L)$. A classification problem would set one component $b_K = 1$ during training, placing the input $\mathbf{a}$ in class K. Each ART_a input is complement coded, with $\mathbf{I} = \mathbf{A} = (\mathbf{a}, \mathbf{a}^c)$, where $0 \le a_i \le 1$ and $a_i^c \equiv 1 - a_i$. Note then that the dimension of the input vector $\mathbf{A}$ equals $2M$ and the city-block norm of $\mathbf{A}$, defined by

$$|\mathbf{A}| \equiv \sum_{i=1}^{2M} A_i,$$

equals M. The output $\mathbf{b}$ is normalized to 1:

$$|\mathbf{b}| \equiv \sum_{k=1}^{L} b_k = 1$$

corresponding to a category probability distribution. During testing, search may occur if the baseline vigilance parameter ($\bar{\rho}$) is positive. In ART_a, each top-down weight w_{ji} is identically equal to the bottom-up weight w_{ij}, and the weight vector $\mathbf{w}_j$ represents both $(w_{1j} \ldots w_{ij} \ldots w_{2M,j})$ and $(w_{j1} \ldots w_{ji} \ldots w_{j,2M})$. Instance counting enumerates the number of times a category is activated during training. With category choice during testing as well as training, instance counting does not affect prediction and the ARTMAP-IC algorithm is equivalent to an ARTMAP algorithm.

3. Match tracking and inconsistent cases

Inconsistent cases, where identical input feature sets correspond to patients with different outcomes, often appear in medical databases. The basic ARTMAP network, run in the

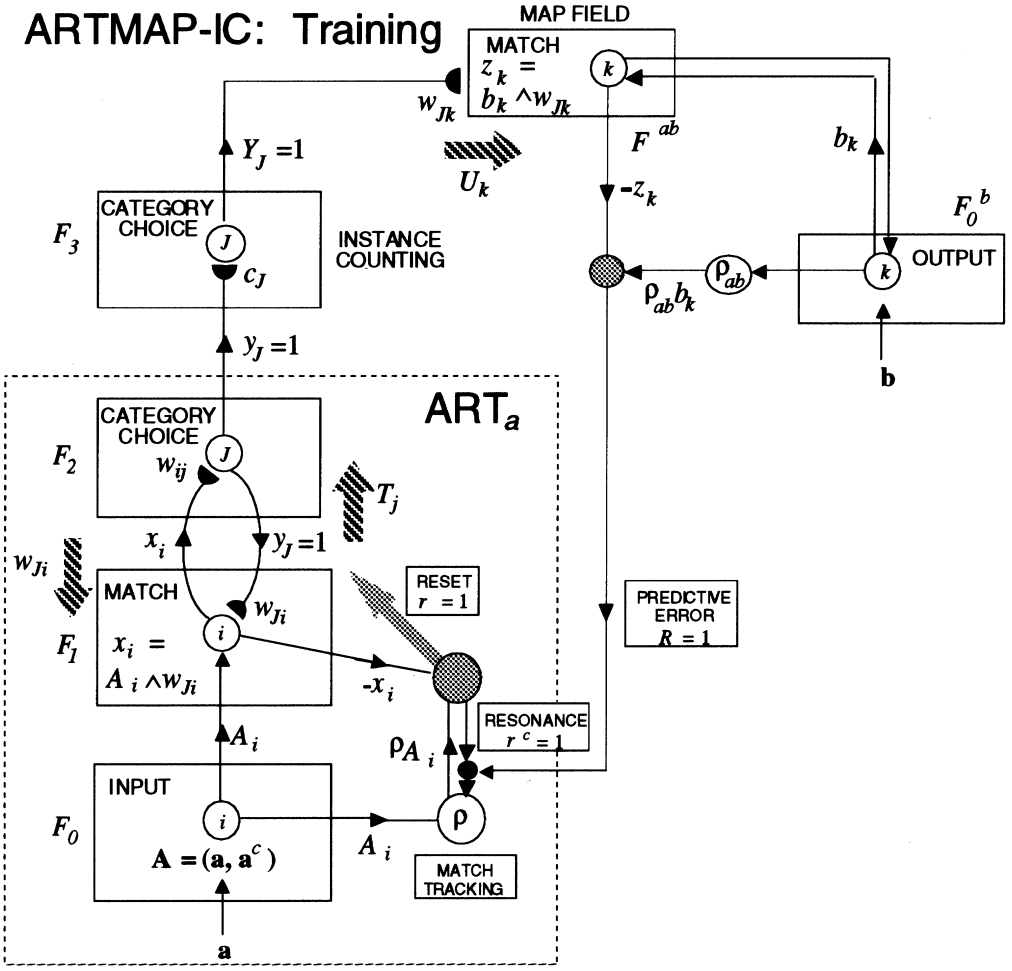

Fig. 2. ARTMAP-IC adds an instance counting layer F_3 to the ARTMAP network. Training is the same as for ARTMAP, except that a counting weight c_j enumerates the number of instances placed in each category j.

fast learning mode, would learn the first such instance and its predicted outcome, but would then be unable to encode the inconsistent cases. Slow learning would average across cases and provide a more probabilistic outcome estimate, but would sacrifice the network's ability to encode rare cases. A small modification of the ARTMAP *match tracking* search algorithm allows the network to encode inconsistent cases and make distributed probability estimates during testing, even when training employs fast learning. In addition, the new algorithm has been found in a number of database examples to compress memory by a factor of 50–100% compared to the original algorithm, without loss of predictive accuracy. Finally, it is actually a better approximation to the real-time ARTMAP network differential equation model, as follows.

In ART models, a *vigilance parameter* ρ establishes a network matching criterion that, if not met, leads to category reset and search. In ARTMAP networks, match tracking is the process that raises the ART_a vigilance ρ to correct predictive errors. Vigilance becomes an internally controlled variable that obeys the differential equation:

$$\frac{\mathrm{d}}{\mathrm{d}t}\rho = -(\rho - \bar{\rho}) + \Gamma R r^c. \tag{1}$$

In Eq. (1), $\bar{\rho}$ is a baseline vigilance parameter, R is a predictive error indicator signal from F^{ab} to ART_a, r/r^c are complementary ART_a reset/resonance indicator signals, and $\Gamma \gg 1$ (Fig. 2). The vigilance relaxation rate is O(1), which is assumed to be slow on the time scale of search and fast on the time scale of learning. Thus, during learning, when $r^c = 1$ and $R = 0$, ρ decreases toward $\bar{\rho}$.

The activity vector $\mathbf{x}$ at F_1 represents a match between the input $\mathbf{A}$ and the $F_2 \rightarrow F_1$ signal, which equals the weight vector $\mathbf{w}_J$ when F_2 makes a choice. When the Jth F_2 node is chosen

$$x_i = A_i \wedge w_{Ji}, \tag{2}$$

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

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

Thus, $\mathbf{x} = \mathbf{A} \wedge \mathbf{w}_J$. Similarly, the F^{ab} activity vector $\mathbf{z}$ represents a match between the output $\mathbf{b}$ and the total signal $\mathbf{U} \equiv (U_1 \ldots U_k \ldots U_L)$ from ART_a to F^{ab}, so

$$z_k = b_k \wedge U_k \tag{4}$$

and $\mathbf{z} = \mathbf{b} \wedge \mathbf{U}$.

In the ARTMAP-IC algorithm (Section 7), the network detects a predictive error when

$$|\mathbf{z}| < \rho_{ab}|\mathbf{b}| = \rho_{ab}, \tag{5}$$

where ρ_{ab} is the map field vigilance parameter. Then $R = r^c = 1$ and ρ begins to rise rapidly, according to Eq. (1). However, as soon as ρ becomes just large enough to satisfy the inequality:

$$|\mathbf{x}| = |\mathbf{A} \wedge \mathbf{w}_J| < \rho|\mathbf{A}| = \rho M \tag{6}$$

the network resets ART_a. While the reset indicator signal ($r = 1$) triggers a search for a new F_2 coding node, the complementary resonance indicator shuts off ($r^c = 0$), halting the rise of ρ, by Eq. (1). A predicted error thus causes vigilance to "track the F_1 match", since ρ increases until it has reached the ART_a match value $|\boldsymbol{A} \wedge \boldsymbol{w}_J||\boldsymbol{A}|^{-1}$.

ART_a search selects a new F_2 node while ρ remains large. A newly active node must thereby meet a stricter matching criterion to establish resonance and maintain stable activity long enough to generate a new map field prediction. The original ARTMAP simulations approximated this process with a match tracking algorithm (MT+) that did not allow ρ to decay *at all* during search, as if the search cycle were infinitely fast. After J is reset, then

$$\rho = |\mathbf{A} \wedge \mathbf{w}_J||\mathbf{A}|^{-1} + \epsilon \tag{7}$$

where $0 < \epsilon \ll 1$. A modified match tracking algorithm (MT−) postulates a rapid but finite search rate, allowing ρ to decay slightly before the next chosen node is tested against the matching criterion. In Eq. (7), then, MT− sets $\epsilon \leq 0$, which allows identical inputs that predict different outcomes to establish distinct recognition categories.

Search ends when the active patterns meet the vigilance matching criterion at ART_a:

$$|\mathbf{x}| \geq \rho|\mathbf{A}| \tag{8}$$

and at the map field:

$$|\mathbf{z}| \geq \rho_{ab}|\mathbf{b}|. \tag{9}$$

With category choice at ART_a, $U_k = w_{Jk}$ for $k = 1 \ldots L$, where J is the chosen node at F_2. Thus, by Eqs. (8) and (9), since $|\mathbf{A}| = M$ and $|\mathbf{b}| = 1$, search ends when:

$$\sum_{i=1}^{2M} A_i \wedge w_{iJ} \geq \rho M \tag{10}$$

and

$$\sum_{k=1}^{L} b_k \wedge w_{Jk} \geq \rho_{ab}. \tag{11}$$

When $\mathbf{b}$ represents a single output class K, $b_K = 1$ so the map field matching criterion (Eq. (11)) reduces to the criterion $w_{JK} \geq \rho_{ab}$.

Setting the baseline vigilance $\bar{\rho} = 0$ maximizes code compression. Setting $\bar{\rho} > 0$ establishes a minimum matching criterion that must be met before a chosen node can make a prediction. Thus, $\bar{\rho}$ can serve as a predictive *confidence threshold*.

4. Voting

ARTMAP fast learning typically produces different adaptive weights and ART_a recognition categories for different orderings of a given training set, even when the overall predictive accuracy of each such trained network is similar. The different category structures cause variations among the

locations of test set errors as training set input orderings vary. A voting strategy uses several ARTMAP systems that are separately trained on one input set with different orderings. The final prediction for a given test set item is based on predictions of networks in a voting "committee". Since the set of items making erroneous predictions varies from one ordering to the next, voting serves both to cancel some of the errors and to assign confidence estimates to competing predictions. A committee of about five to ten voters has proved suitable in many examples, and the marginal benefits of voting are most apparent when the number of training samples is limited.

5. ART-EMAP distributed prediction by the *Q*-max rule

To improve performance in a noisy or ambiguous input environment, ART-EMAP adds spatial and temporal evidence accumulation processes to the basic ARTMAP system (Carpenter & Ross, 1993, 1995). ART-EMAP (Stage 1) distributes activity across category representations during performance. In a variety of studies, this device improves test-set predictive accuracy compared to ARTMAP, which is the same network with category choice during testing. Distributed test-set category activation also improves performance accuracy on the medical database simulations below (Section 8). Further improvement is achieved by the addition of an instance counting measure (Section 6) that weights distributed predictions according to the number of training set inputs placed in each category.

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

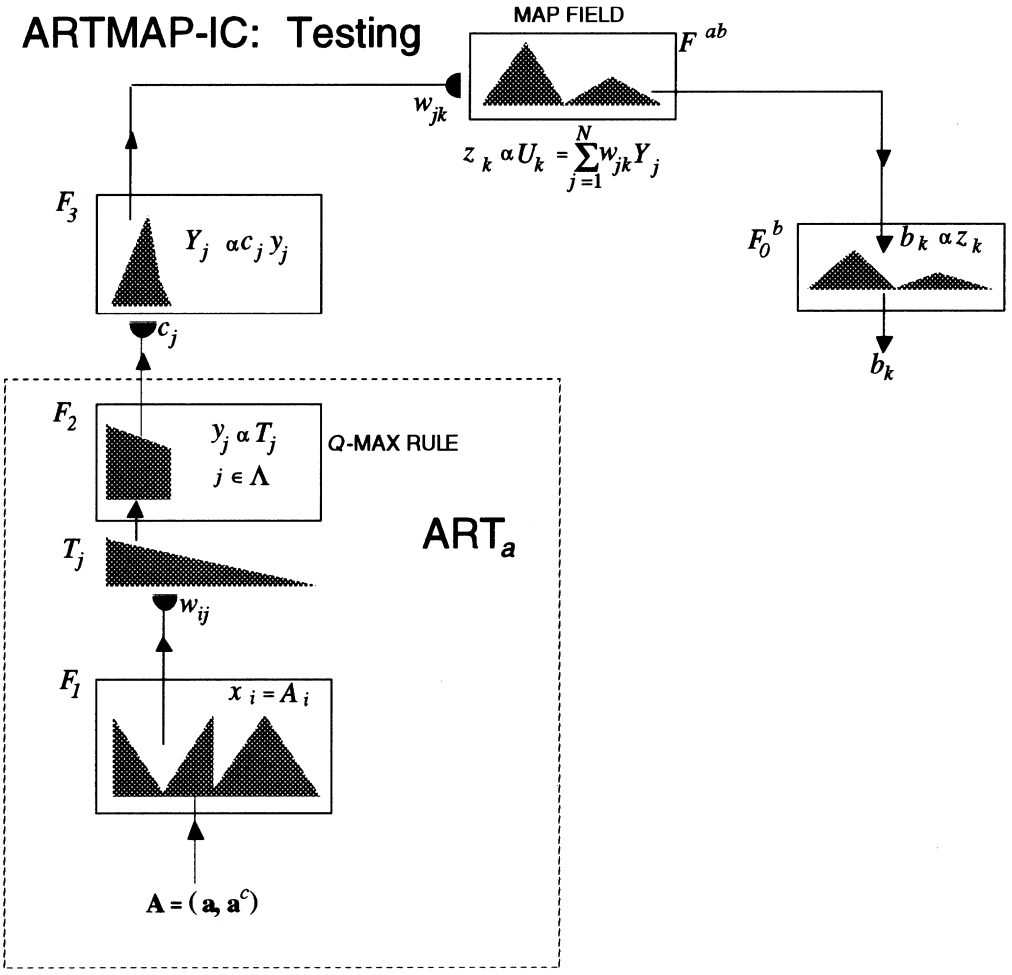

Fig. 3. During testing, an input activates Q category nodes, in proportion to the input from F_1 to the category field F_2. After multiplication by the instance counting weights to produce distributed activation Y_j at F_3, the Q active nodes project to the map field F^{ab} via the map field weights w_{jk} to form a distributed prediction vector **U**. The network then computes classification probabilities, with $|\mathbf{b}| = 1$ at an output field F_0^b.

proportional to T_j. That is,

$$Q-\text{max rule}: \quad y_j = \begin{cases} \dfrac{T_j}{\sum_{\lambda\in\Lambda} T_\lambda} \text{ if } j \in \Lambda \\ 0 \text{ if } j \notin \Lambda \end{cases} \tag{12}$$

where Λ is the set of Q nodes with the largest T_j values (Fig. 3). The way a Q-max rule makes test set predictions is analogous to a K nearest neighbor (KNN) algorithm with $K = Q$. When $Q = 1$, the Q-max rule reduces to category choice. In the simulations below both ART-EMAP and ARTMAP-IC use the Q-max rule during testing.

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

$$\text{Rule}-\text{of}-\text{thumb } Q \text{ value}: \quad Q = \min\left\{\frac{C}{2L}, 30\right\}. \tag{13}$$

This estimate requires no separate verification step and gives good results on the four sets of medical database simulations (Section 8), where the number of output classes is $L = 2$, corresponding to good or bad outcomes. In the end, test set results can also be examined over a range of Q values to check for parameter sensitivity.

6. Instance counting

Instance counting biases distributed predictions according to the number of training set inputs classified by each F_2 node. Fig. 3 illustrates how an ARTMAP network with an extra field F_3 can implement instance counting. During testing the $F_2 \rightarrow F_3$ input y_j is multiplied by the counting weight c_j to produce normalized F_3 activity Y_j, which projects to the map field F^{ab} for prediction. That is, for $j = 1,\ldots,N$, activity at the counting field F_3 is:

$$Y_j = \frac{c_j y_j}{\sum_{\eta=1}^{N} c_\eta y_\eta}. \tag{14}$$

The input U_k from F_3 to the kth map field node is then:

$$U_k = \sum_{j=1}^{N} w_{jk} Y_j = \frac{\sum_{j=1}^{N} w_{jk} c_j y_j}{\sum_{j=1}^{N} c_j y_j} \tag{15}$$

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

$$Y_j = y_j = \begin{cases} 1 \text{ if } j = J \\ 0 \text{ if } j \neq J \end{cases} \tag{16}$$

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

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

$$c_j^{(\text{new})} = c_j^{(\text{old})} + y_j \tag{17}$$

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

During testing (Section 7.2), when distributed F_2 activation is determined by a Q-max rule (Eq. (12)), the map field input is

$$U_k = \frac{\sum_{j=1}^{N} w_{jk} c_j y_j}{\sum_{j=1}^{N} c_j y_j} = \frac{\sum_{j\in\Lambda} w_{jk} c_j T_j}{\sum_{j\in\Lambda} c_j T_j} \tag{18}$$

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

7. ARTMAP-IC algorithm

The algorithms below summarize ARTMAP-IC dynamics during training (Section 7.1) and testing (Section 7.2). During training, ART_a makes a category choice. During testing, a distributed ART_a category representation generates an output class probability vector $\mathbf{b}$. When ART_a makes a choice during testing ($Q = 1$), the ARTMAP-IC algorithm is equivalent to a fuzzy ARTMAP algorithm. However, the original ARTMAP notation has been changed somewhat to clarify network functions and for consistency with a family of more general ART systems (Carpenter, 1997).

7.1. ARTMAP-IC training algorithm

During training, input–output pairs $(\mathbf{a}^{(1)}, \mathbf{b}^{(1)})$, $(\mathbf{a}^{(2)}, \mathbf{b}^{(2)}), \ldots, (\mathbf{a}^{(n)}, \mathbf{b}^{(n)}), \ldots$ are presented for equal time intervals. With complement coding, the ART_a input $\mathbf{A} = (\mathbf{a}, \mathbf{a}^c)$. Voting would repeat the training procedure several times, each with a different ordering of the input–output sequence.

1. Variables: $i = 1\ldots2M$, $j = 1\ldots N$, $k = 1\ldots L$

Activation	Weights	$F_1 \to F_2$ signals	$F_3 \to F^{ab}$ signal
x_i—F_1 (matching)	w_{ij}—$F_1 \leftrightarrow F_2$	S_j—Phasic	U_k—Total
y_j—F_2 (coding)	c_j—$F_2 \to F_3$	Θ_j—Tonic	ρ—ART_a vigilance
Y_j—F_3 (counting)	w_{jk}—$F_3 \to F^{ab}$	T_j—Total	C—no. of committed nodes
z_k—F^{ab} (map field)			

2. Notation

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

3. Signal rule: Define the $F_1 \to F_2$ signal function $T_j = g(S_j, \Theta_j)$, where $g(0,0) = 0$ and

$$\frac{\partial g}{\partial S_j} > \frac{\partial g}{\partial \Theta_j} > 0$$

for $S_j > 0$ and $\Theta_j > 0$.
E.g. $T_j = S_j + (1-\alpha)\Theta_j$ with $\alpha \in (0,1)$ (choice-by-difference) or $T_j = S_j/(\alpha + 2M - \Theta_j)$ with $\alpha > 0$ (Weber law). In ARTMAP, ART-EMAP, and ARTMAP-IC, the *phasic signal* component S_j is defined by

$$S_j = \sum_{i=1}^{2M} A_i \wedge w_{ij}$$

and the *tonic signal* component Θ_j is defined by

$$\Theta_j = \sum_{i=1}^{2M} (1 - w_{ij}).$$

E.g. $T_j = |\mathbf{A} \wedge \mathbf{w}_j| + (1-\alpha)(2M - |\mathbf{w}_j|)$ with $\alpha \in (0,1)$ (choice-by-difference) or

$$T_j = \frac{|\mathbf{A} \wedge \mathbf{w}_j|}{\alpha + |\mathbf{w}_j|}$$

with $\alpha > 0$ (Weber law).

4. Parameters

   Number of input components—$i = 1\ldots2M$
   Number of coding nodes—$j = 1\ldots N$
   Number of output components—$k = 1\ldots L$
   Signal rule parameters—e.g. $\alpha \in (0,1)$ (choice-by-difference) or $\alpha > 0$ (Weber law)
   Learning rate—$\beta \in [0,1]$, with $\beta = 1$ for fast learning
   Baseline vigilance (ART_a)—$\bar{\rho} \in [0,1]$, with $\bar{\rho} = 0$ for maximal code compression
   Map field vigilance—$\rho_{ab} \in [0,1]$, with $\rho_{ab} \cong 1$ for maximal output separation
   Match tracking—ϵ, with $|\epsilon|$ small.

   MT +: $\epsilon > 0$
   MT −: $\epsilon \le 0$

   F_2 order constants—$0 < \Phi_N < \ldots < \Phi_j < \ldots < \Phi_1 < g(M,0)$, with all $\Phi_j \cong g(M,0)$.

5. First iteration: $n = 1$

   $F_1 \leftrightarrow F_2$ ART_a weights—$w_{ij} = 1$, $i = 1\ldots2M$, $j = 1\ldots N$
   $F_2 \to F_3$ counting weights—$c_j = 0$, $j = 1\ldots N$
   $F_3 \to F^{ab}$ map field weights—$w_{jk} = 1$, $j = 1\ldots N$, $k = 1\ldots L$
   Number of committed nodes—$C = 0$
   Signal to uncommitted nodes—$T_j = \Phi_j$, $j = 1\ldots N$

   ART_a vigilance—$\rho = \bar{\rho}$
   Input:

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

   Output:

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

6. Reset: New steady state at F_2 and F_1

   Choose a category—Let J be the index of the F_2 node with maximal input T_j, i.e. $T_J = \max\{T_1\ldots T_N\}$
   Number of committed nodes—If $J > C$, set $C = J$
   F_1 activation—$x_i = A_i \wedge w_{iJ}$ $\quad i = 1\ldots2M$

7. Refractory signal: $F_1 \to F_2$ signal is deactivated on the time scale of search

   $T_J = 0$

8. Reset or prediction: Check the F_1 matching criterion

   If

$$\sum_{i=1}^{2M} x_i < \rho M$$

go to (6) Reset
If

$$\sum_{i=1}^{2M} x_i \geq \rho M$$

go to (9) Prediction

9. Prediction:

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

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

If

$$\sum_{k=1}^{L} z_k < \rho_{ab}$$

go to (11) Match tracking
If

$$\sum_{k=1}^{L} z_k \geq \rho_{ab}$$

go to (12) Resonance

11. Match tracking: Raise ρ to the point of ART_a reset

$$\rho \frac{=_1}{M} \sum_{i=1}^{2M} x_i + \epsilon$$

Go to (6) Reset

12. Resonance: New weights on the time scale of learning

Old weights—$w_{iJ}^{\mathrm{old}} = w_{iJ}$ $i = 1\ldots 2M$

$c_J^{\mathrm{old}} = c_J$
$w_{Jk}^{\mathrm{old}} = w_{Jk}$ $k = 1\ldots L$

Decrease $F_1 \rightarrow F_2$ weights—$w_{iJ} = (1-\beta)w_{iJ}^{\mathrm{old}} + \beta(A_i \wedge w_{iJ}^{\mathrm{old}})$ $i = 1\ldots 2M$
Increase $F_2 \rightarrow F_3$ counting weight—$c_J^{\mathrm{old}} = c_J + 1$
Decrease $F_2 \rightarrow F^{ab}$ weights—$w_{Jk} = (1-\beta)w_{Jk}^{\mathrm{old}} + \beta(b_k \wedge w_{Jk}^{\mathrm{old}})$ $k = 1\ldots L$
ART_a vigilance recovery—$\rho = \bar{\rho}$

13. Next iteration: Increase n by 1

New input:

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

New output: $b_k = b_k^{(n)}$ $k = 1\ldots L$
New F_1 activation: $x_i = A_i \wedge w_{iJ}$ $i = 1\ldots 2M$
New $F_1 \rightarrow F_2$ signal to committed nodes

Phasic:

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

Tonic:

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

Total:

$$T_j = g(S_j, \Theta_j) \quad j = 1\ldots C$$

Go to (6) Reset

7.2. ARTMAP-IC testing

During ARTMAP-IC testing, $F_1 \leftrightarrow F_2$ categorization weights w_{ij}, $F_2 \rightarrow F_3$ counting weights c_j, and $F_3 \rightarrow F^{ab}$ prediction weights w_{jk} are fixed, and the baseline vigilance parameter $\bar{\rho} = 0$, so no search occurs. A test-set input **a** activates a distributed category representation at ART_a, by the Q-max rule, where Q is a fixed number of F_2 nodes. Setting $Q = 1$ reduces ARTMAP-IC to an ARTMAP algorithm with category choice and setting $Q = N$ engages the entire trained system in the net prediction. Filtered through the instance counting weights c_j and the map field weights w_{jk}, the distributed category representation produces a normalized distributed output probability vector **b**.

In the medical database problems in Section 8, the output **b** represents two classes corresponding to good ($k = 1$) or bad ($k = 2$) outcomes. With two such classes, the prediction "bad" could be made whenever $b_2 \geq 0.5$. However, instance counting tends to weigh against rare cases, which often correspond to bad outcomes. To offset this bias, a good/bad decision threshold τ may be set below 0.5, with a "bad" prediction whenever $b_2 \geq \tau$. In all four sets of ARTMAP-IC simulations below, $\tau = 0.4$.

For voting, the network generates a set of prediction vectors for each of the trained networks produced by several different orderings of the training set inputs. The voting networks may average their output vectors **b** for each input **a** or each voting network may choose one output class, with the predicted class being the one that receives the most votes. Simulations in Section 8 employ the former voting method.

1. Test set input

Input:

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

2. $F_1 \rightarrow F_2$ signal

Phasic:

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

Tonic:

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

Total:

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

3. F_2 activation by the Q-max rule:

Let Λ be the index set of the Q F_2 node with maximal input T_j.
That is, $\Lambda \subseteq \{1 \ldots N\}$, $|\Lambda| = Q$, and $T_J \geq T_j$ for $J \in \Lambda$ and $j \notin \Lambda$.
Setting $Q = 1$ gives choice, or winner-take-all, activation at F_2.

4. Output prediction:

$$b_k = \frac{\sum_{j \in \Lambda} w_{jk} c_j T_j}{\sum_{\kappa=1}^{L} \sum_{j \in \Lambda} w_{j\kappa} c_j T_j} \quad k = 1 \ldots L$$

8. Comparative simulations

Benchmark medical database studies examine the benefits of distributed prediction and instance counting in the ARTMAP-IC network. ARTMAP-IC performance is compared to that of the basic ARTMAP network, with category choice, and ART-EMAP, which uses distributed category prediction but not instance counting. The various ARTMAP networks are also compared with logistic regression, ADAP, and KNN on a Pima Indian diabetes database (Section 8.1); with logistic regression, a multisurface method of pattern separation, and KNN on a breast cancer database (Section 8.2); and with logistic regression, CLASSIT, instance-based (IBL) classifiers, C4, and KNN on a heart disease database (Section 8.3). In nearly every case, ARTMAP-IC with instance counting has the best performance statistics. A fourth study shows how the modified match tracking algorithm MT $-$ combines with Q-max distributed prediction and instance counting to allow ARTMAP-IC to encode inconsistent cases. On this gall bladder removal (cholecystectomy) database, ARTMAP-IC performance is just above that of logistic regression and better than ART-EMAP, KNN, and basic fuzzy ARTMAP (Section 8.4).

Table 1 shows the basic ARTMAP, ART-EMAP, and ARTMAP-IC network simulation parameters and the instance counting and match tracking rules and Table 2 compares database characteristics. A preliminary study led to network parameter estimates, then the ARTMAP system definition was held constant across all simulations and all four databases. The Pima Indian diabetes study uses the same training and testing sets as in the benchmark ADAP simulations and the heart disease study uses the same training and testing sets as in the benchmark IBL simulations. The other two studies use five-fold cross validation (Mosier, 1951) which divides the input set into five parts, each of which serves, in turn, as a test set, with average results reported. In all ART-EMAP and KNN simulations, the system predicts whichever of the two outcomes k (good or bad) receives the larger net input U_k from ART_a at the map field F^{ab}. Since ARTMAP-IC reduces the influence of rare cases, which usually represent bad outcomes, a large majority of evidence for a bad outcome was considered noteworthy enough to adjust the decision boundary somewhat toward this prediction. Thus the network predicts a bad outcome when the net input to the corresponding node is at least 40% of the total input to F^{ab}. These decision thresholds (0.4 for ARTMAP-IC and 0.5 for all other systems) are held constant across the four sets of studies. All ARTMAP results reflect the participation of ten voters.

Simulation results report the C-index (Harrell et al., 1984, 1985) as well as the correct prediction rate. The C-index is a measure of predictive score that is independent of both the mixture of good/bad test set cases and the bad-case decision threshold. In an ARTMAP network, the C-index measures the probability that, for any randomly selected pair of bad/good test set cases, the signal sent by the bad case to the "bad" map field node will be larger than the signal sent by the good case to that node. The C-index is equivalent to the area under the Receiver Operating Characteristic (ROC) curve, which plots the true positive (bad case) prediction

Table 1
ARTMAP, ART-EMAP, and ARTMAP-IC simulation parameters

Choice parameter	$\alpha = 0.1$	
Learning rate parameter	$\beta = 1.0$	
Baseline vigilance	$\bar{\rho} = 0.0$	
Bad-case decision threshold	$\tau = 0.5$	ARTMAP, ART-EMAP, KNN
	$\tau = 0.4$	ARTMAP-IC
Signal rule	$T_j = S_j/(\alpha + 2M - \Theta_j)$ (Weber law)	
F_2 order constants	$0 < \Phi_N < \ldots < \Phi_j < \ldots < \Phi_1 < g(M, 0)$ with $\Phi_j \cong g(M,0)$	
Number of voters	10	

Table 2
Database characteristics

Data set	No. training set inputs	% bad outcomes	No. input components (M)	Match tracking rule	No. ARTMAP categories (C) average [range]	Rule-of-thumb Q values
Diabetes	576	34.9	8	MT + $\epsilon = +0.0001$	62 [50–74]	15 [12–19]
				MT − $\epsilon = -0.0001$	62 [53–68]	15 [13–17]
				MT − $\epsilon = -0.01$	45 [31–54]	11 [8–14]
Breast cancer	559	34.5	9	MT + $\epsilon = +0.0001$	14 [8–20]	3–4 [2–5]
Heart disease	250	45.9	13	MT + $\epsilon = +0.0001$	26 [20–33]	6 [5–8]
Gall bladder	2546	16.4	16	MT − $\epsilon = -0.0001$	450 [375–594]	30
				MT − $\epsilon = -0.01$	286 [209–335]	30

rate against the false positive rate for a given test. Logistic regression simulations use the standard SAS PROC Logistic statistical package (SAS Institute, 1990).

8.1. Pima Indian Diabetes database

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

Smith et al. (1988) used the PID data set to evaluate the preceptron-like ADAPtive learning routine (ADAP). This study had 576 cases in the training set and 192 cases in the test set, and comparative simulations in this section all use the same training and test sets. About 34.9% of patients in the sample developed diabetes. Table 3 compares ADAP test set performance with that of logistic regression, KNN, and three ARTMAP networks. ARTMAP-IC uses the instance counting (IC) rule (Eq. (17)) and the Q-max rule (Eq. (12)) for distributed prediction. Comparative simulations show results for ART-EMAP (Stage 1), which is equivalent to ARTMAP-IC without instance counting; and

Table 3
Pima Indian Diabetes (PID) simulation results

Model	Correct predictions	C-index	Compression factor	
Logistic regression	77%	0.84	–	
ADAP	76%	–	–	
ARTMAP ($Q = 1$) [MT + : $\epsilon = +0.0001$]	66%	0.76	9.3	
	$Q = 15$	$12 \le Q \le 19$	Peak % [C-index, Q]	Compression
KNN	77%	76–77%	77% [0.80, $Q = 13$–15]	1
ART-EMAP [MT + : $\epsilon = +0.0001$]	76%	76–78%	78% [0.87, $Q = 13$]	9.3
ARTMAP-IC [MT + : $\epsilon = +0.0001$]	79%	79–80%	80% [0.87, $Q = 9$–13]	9.3
	$Q = 15$	$13 \le Q \le 17$		
ARTMAP-IC [MT − : $\epsilon = -0.0001$]	81%	80–81%	81% [0.88, $Q = 15$]	9.3
	$Q = 11$	$8 \le Q \le 14$		
ARTMAP-IC [MT − : $\epsilon = -0.01$]	79%	78–81%	81% [0.87, $Q = 9$]	12.8

for basic ARTMAP, which sets $Q = 1$ for category choice during testing. With the original match tracking rule MT+, the various ARTMAP networks share a common training regime. On average, these networks produced 62 committed category nodes ($C = 62$), with this number ranging from $C = 50$ to $C = 74$ across simulations, depending on training set input presentation order (Table 2). The average and range of values of C provide a priori rule-of-thumb estimates (Eq. (13)) for the number of active nodes Q in a distributed category representation. With two output classes ($L = 2$), the target value is $Q = 15$, with an expected range of values from $Q = 12$ to $Q = 19$.

Table 3 shows that the basic ARTMAP network ($Q = 1$) does not perform well on the PID database problem, but that the same trained network with distributed test set prediction (ART-EMAP) brings performance up to the level of logistic regression, ADAP, and KNN. Instance counting (ARTMAP-IC) improves performance even further, both in terms of the C-index and the number of correct test set predictions. Table 3 shows that the rule-of-thumb estimate identifies Q values that are nearly optimal, and that performance is robust across a range of Q values. Compared to KNN, ARTMAP networks with $\epsilon = \pm 0.0001$ compress memory by a factor of 9.3:1. Although the PID database has no inconsistent cases, the MT− match tracking rule ($\epsilon = -0.01$) compresses memory even more than the same network with $\epsilon = \pm 0.0001$, reducing the number of committed nodes from $C = 62$ to $C = 45$, with no deterioration in predictive accuracy.

8.2. Breast cancer database

The University of Wisconsin breast cancer database (Wolberg & Mangasarian, 1990) provides laboratory data from 699 patients with tumors, of which 458 (65.5%) proved to be benign and 241 (34.5%) malignant. A patient record from a breast fine-needle aspirate lists nine cytological characteristics: clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, and mitoses. The data set labels each cytological characteristic from 1 (benign) to 10 (malignant), although no one characteristic was considered a reliable predictor.

Wolberg and Mangasarian (1990) applied a multisurface method of pattern separation, training on 246 of the 369 inputs available at that time to obtain 96% test set predictive accuracy. Training on 80% of the current data set (559 inputs) for five-fold cross validation, the ARTMAP, KNN, and logistic regression classifiers performed comparably well (Table 4). Compared to KNN, the ARTMAP networks compressed the training set by a factor of 40, storing from eight to 20 category nodes, with an average of 14, for each simulation. As in the PID database, the rule-of-thumb estimate (Eq. (13)) provided a good Q value. In this case, where all classifiers seem to reach near-optimal performance levels, instance counting provides no marginal benefits.

8.3. Heart disease database

The Cleveland heart disease database from the UCI repository (Murphy & Aha, 1992), was gathered from 303 cardiology patients at the Long Beach V.A. Medical Center and the Cleveland Clinic Foundation. Each record stores 13 attributes: age, sex, chest pain type, blood pressure, cholesterol level, fasting blood sugar, resting electrocardiograph results, maximum heart rate, angina, ST depression induced by exercise relative to rest, the slope of the peak exercise ST segment, the number of major vessels colored by fluoroscopy, and thalassemia. Six patient records have many missing values. In the current simulations (logistic regression, KNN, and all ARTMAP systems), values of the missing components are set to 0, which denotes a normal attribute value. Of the 303 patients, 164 (54.1%) were diagnosed as healthy and 139 (45.9%) as having heart disease, defined as blood vessels narrowed by more than 50%. The database author, R. Detrano, estimates that the class labels have an error rate of about 20%.

Benchmark studies of the heart disease database apply the unsupervised CLASSIT algorithm (Gennari et al., 1989); instance-based (IBL) classifiers, which are similar to KNN

Table 4
Breast cancer simulation results

Model	Correct predictions	C-index	Compression factor	
Logistic regression	97%	0.993	–	
Multisurface pattern separation (trained on 234 inputs)	96%	–	–	
ARTMAP ($Q = 1$) [MT+: $\epsilon = +0.0001$]	96%	0.987	40	
	$Q = 3$–4	$2 \le Q \le 5$	Peak % [C-index, Q]	Compression
KNN	96%	96%	97% [0.958, $Q = 1$]	1
ART-EMAP [MT+: $\epsilon = +0.0001$]	97%	97%	97% [0.994, $Q = 3$]	40
ARTMAP-IC [MT+: $\epsilon = +0.0001$]	96%	96%	96% [0.992, $Q = 3$]	40

Table 5
Heart disease simulation results

Model	Correct predictions	C-index	Compression factor	
Logistic regression	79.0%	0.88	–	
CLASSIT	78.9%	–	101	
IB1	75.7 ± 0.8%	–	1	
IB2	71.4 ± 0.8%	–	3.3	
IB3	78.0 ± 0.8%	–	13	
C4	75.5 ± 0.7%	–	–	
ARTMAP ($Q = 1$) [MT + : $\epsilon = +0.0001$]	74%	0.84	9.6	
	$Q = 6$	$5 \leq Q \leq 8$	Peak % [C-index, Q]	Compression
KNN	67%	66–68%	68% [0.69, $Q = 5$]	1
ART-EMAP [MT + : $\epsilon = +0.0001$]	76%	75–76%	77% [0.84, $Q = 9$–11]	9.6
ARTMAP-IC [MT + : $\epsilon = +0.0001$]	78%	78%	81% [0.84, $Q = 19$]	9.6

(Aha et al., 1991); and the decision tree algorithm C4 (Quinlan, 1986). Simulations here used the same training set of 250 inputs as in the benchmark studies. Table 5 shows that KNN does not perform well on this problem and that logistic regression, CLASSIT, IB3, and ARTMAP-IC perform near the estimated optimal level of 80% correct prediction rate.

8.4. Cholecystectomy (gall bladder removal) database

The cholecystectomy database represents 3182 randomly selected Medicare patients from seven states. The prediction task is to estimate the likelihood of an adverse event, defined as the occurrence of at least one of 16 possible types of severe complications. Adverse events occurred in 16.4% of the cases. Each input was derived from 62 features recorded from pre-admission testing, admission history, and laboratory and procedure results. Preprocessing reduced the number of input components to 16 after features significantly associated ($p < 0.5$) with each of the 16 types of adverse events were merged.

The cholecystectomy database contains 59 pairs of inconsistent data vectors. That is, for each pair, identical inputs predict opposite outcomes. The database can thus be used to examine the effect of the new ARTMAP match tracking algorithm, MT − . Recall that MT − allows a network to learn from inconsistent cases during training (Section 3). During testing, then, distributed category activation, with instance counting, can provide a likelihood estimate of an adverse event that benefits from the knowledge of inconsistent training set pairs. To train basic ARTMAP and ART-EMAP with the original MT + algorithm, inconsistent inputs were recast by small random perturbations. Even with the recast data, ARTMAP can still choose only the one maximally activated category during testing. ART-EMAP prediction would reflect the competing category predictions, but would not reflect the number of training set instances coded by each category.

Table 6 compares the C-index performance measures for logistic regression and basic ARTMAP with those of KNN, ART-EMAP, and ARTMAP-IC. The overall predicted accuracy of all the classifiers is low, but differences between

Table 6
Gall bladder removal (cholecystectomy) results

Model	C-index	Compression factor	
Logistic regression	0.68	–	
ARTMAP ($Q = 1$) [MT + : $\epsilon = +0.0001$]	0.63	5.7	
	$Q = 30$	Peak C-index, [Q]	Compression
KNN	0.65	0.67, [$Q = 55$–60]	1
ART-EMAP [MT + : $\epsilon = +0.0001$]	0.66	0.66, [$Q = 22$–58]	5.7
ARTMAP-IC [MT − : $\epsilon = -0.0001$]	0.69	0.69, [$Q = 9$–35]	5.7
ARTMAP-IC [MT − : $\epsilon = -0.01$]	0.68	0.69, [$Q = 5$–9]	8.9

classifiers are still apparent. With distributed prediction, ART-EMAP and ARTMAP-IC perform consistently better than ARTMAP with category choice. The ARTMAP networks with $\epsilon \pm 0.0001$ create about 450 categories during training and so compress the input data by about 5.7:1. KNN does not compress the data, and the algorithm does not perform well until Q exceeds 20. After that it is comparable to ART-EMAP with the Q-max category activation rule. For Q greater than 10, ARTMAP-IC consistently outperforms logistic regression by a small margin.

With $\epsilon = -0.0001$, MT− does not increase code compression compared to MT+. However, decreasing ϵ to -0.01 during training allows MT − to search a large number of nearby categories following a predictive error. This reduces the number of committed nodes from $C = 450$ to $C = 286$, thus increasing the compression ratio from 5.7:1 to 8.9:1, with little effect on performance. Similarly, on the PID data set (Table 3), which has no inconsistent inputs, the MT+ and MT− rules with $\epsilon = \pm 0.0001$ have similar performance rates and numbers of learned categories, while decreasing ϵ to -001 reduces the number of learned categories from 62 to 45.

9. Conclusion

This study provides a self-contained description of ART-MAP neural networks in the context of medical database prediction problems. Instance counting and a modified match tracking algorithm, new components of the ART-MAP family of networks, are introduced and used in combination with ART-EMAP distributed test set prediction. The enhanced ARTMAP networks perform better than the basic ARTMAP system, which uses category choice during both training and testing, and performs as well as or better than a variety of methods applied to benchmark medical prediction problems.

Acknowledgements

This research was supported in part by the National Science Foundation (NSF IRI 94-01659) and the Office of Naval Research (ONR N00014-95-1-0409 ad ONR N00014-95-0657). The authors would like to thank William H. Wolberg from the University of Wisconsin Hospitals, Madison, for providing the breast cancer database; Robert Detrano from the Long Beach V. A. Center for the heart disease database; and Arlene S. Ash, Stephan A. Gaehde, and Mark A. Moskowitz for the cholecystectomy database.

References

Aha, D. W., Kibler, D., & Albert, M. K. (1991). Instance-based algorithms. *Machine Learning, 6,* 37–66.

Baxt, W. G. (1991). Use of an artificial neural network for the diagnosis of myocardial infraction. *Annals of Internal Medicine*, *115*, 843–848.

Bohren, B. F., Hadzikadic, M., & Hanley, E. N. Jr. (1995). Extracting knowledge from large medical databases: An automated approach. *Computers and Biomedical Research*, *28*, 191–210.

Carpenter, G. A. (1997). Distributed learning, recognition, and prediction by ART and ARTMAP neural networks. *Neural Networks*, *10*, 1473–1494. Technical Report CAS/CNS-TR-96-004, Boston, MA: Boston University.

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

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

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

Carpenter, G. A., Grossberg, S., & Reynolds, J. H. (1991a). ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. *Neural Networks*, *4*, 565–588.

Carpenter, G. A., Grossberg, S., & Rosen, D. B. (1991b). Fuzzy ART: Fast stable learning and categorization of analog patterns by an Adaptive Resonance system. *Neural Networks*, *4*, 759–771.

Carpenter, G. A., & Ross, W. D. (1993). ART-EMAP: A neural network architecture for learning and prediction by evidence accumulation. *Proceedings of the World Congress on Neural Networks (WCNN-93)*, Hillsdale, NJ: Lawrence Erlbaum Associates, **III**, 649–656.

Carpenter, G. A., & Ross, W. D. (1995). ART-EMAP: A neural network architecture for object recognition by evidence accumulation. *IEEE Transactions on Neural Networks*, *6*, 805–818.

Downes, P. (1994). Neural network recognition of multiple mammographic lesions. *Proceedings of the World Congress on Neural Networks (WCNN-94)*, Hillsdale, NJ: Lawrence Erlbaum Associates, **I**, 133–137.

Duda, R. O., & Hart, P. E. (1973). *Pattern classification and scene analysis*. New York: Wiley.

Floyd, C. E., Jr., Yun, A. J., Lo, J. Y., Toourassi, G., Sullivan, D. C., & Kornguth, P. J. (1994). Prediction of breast cancer malignancy for difficult cases using an artificial neural network. *Proceedings of the World Congress on Neural Networks (WCNN-94)*, Hillsdale, NJ: Lawrence Erlbaum Associates, **I**, 127–132.

Gennari, J. H., Langley, P., & Fisher, D. (1989). Models of incremental concept formation. *Artificial Intelligence*, *40*, 11–60.

Harrell, F. E. Jr., Lee, K. L., Califf, R. M., Pryor, D. B., & Rosati, R. A. (1984). Regression modeling strategies for improved prognostic prediction. *Statistics in Medicine*, *3*, 43–52.

Harrell, F. E. Jr., Lee, K. L., Matchar, D. B., & Reichert, T. A. (1985). Regression models for prognostic prediction: Advantages, problems, and suggested solutions. *Cancer Treatment Reports*, *69*, 1071–1077.

Howell, O. C. (1992). *Statistical methods for psychology*. Belmont, CA: Duxbury Press.

Lo, S.-C. B., Chan, H.-P., Lin, J.-S., Li, H., Freedman, M. T., & Mun, S. K. (1995). Artificial convolution neural network for medical image pattern recognition. *Neural Networks*, *8*, 1201–1214.

Mangasarian, O. L. (1968). Multi-surface method of pattern separation. *IEEE Transactions on Information Theory*, **IT-14**, 801–807.

Mosier, C. I. (1951). Symposium: The need and the means for cross-validation. 1. Problem and designs of cross-validation. *Education and Psychological Measurement*, *11*, 5–11.

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

Petrick, N., Chan, H., Sahiner, B., Wei, D., Helvie, M. A., Goodsitt, M. M., & Adler, D. D. (1995). Automated detection of breast masses on digital mammograms using a convolution neural network for morphological and texture classification. *Proceedings of the World Congress on Neural Networks (WCNN-95)*, Hillsdale, NJ: Lawrence Erlbaum Associates, **II**, 872–875.

Quinlan, J. R. (1986). The effect of noise on concept learning. In R. S. Michalski, J. C. Carbonell, & T. Mitchell (Eds.) *Machine learning: an artificial intelligence approach*, Vol. II. San Mateo, CA: Morgan Kaufmann Publishers, pp. 149–166.

Rosenberg, C., Erel, J., & Altan, H. (1993). A neural network that learns to interpret myocardial planar thallium scintigrams. *Neural Computation, 5*, 492–501.

Sahiner, B., Chan, H., Petrick, N., Wei, D., Helvie, M. A., Adler, D. D., & Goodsitt, M. M. (1995). Classification of mass and normal breast tissue: An artificial neural network with morphological features. *Proceedings of the World Congress on Neural Networks (WCNN-95)*, Hillsdale, NJ: Lawrence Erlbaum Associates, **II**, 876–879.

SAS Institute (1990). SAS User's Guide, ANOVE-FREQ, Version 6.0. Cary, NC: SAS Institute, Inc.

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

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

Wolbert, W. H., & Mangasarian, O. L. (1990). Multisurface method of pattern separation for medical diagnosis applied to breast cytology. *Proceedings of the National Academy of Sciences of the USA*, **87**, 9193–9196.

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