# II.4

## A NEURAL NETWORK ARCHITECTURE FOR FAST ON-LINE SUPERVISED LEARNING AND PATTERN RECOGNITION

GAIL A. CARPENTER
STEPHEN GROSSBERG
JOHN REYNOLDS

*Center for Adaptive Systems and*
*Graduate Program in Cognitive & Neural Systems*
*Boston University*
*111 Cummington Street*
*Boston, MA 02215*

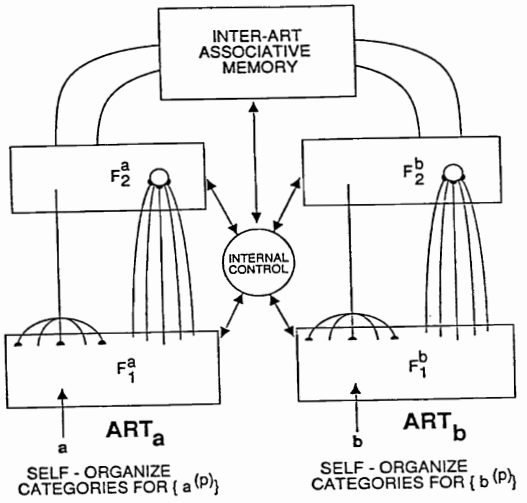
## I. Introduction

This chapter describes a new neural network architecture, called ARTMAP (Carpenter, Grossberg, and Reynolds, 1991), that autonomously learns to classify arbitrarily many, arbitrarily ordered vectors into recognition categories based on predictive success. This supervised learning system is built up from a pair of Adaptive Resonance Theory (Carpenter and Grossberg, 1987a, 1987b, 1988, 1990) modules ($ART_a$ and $ART_b$) that are capable of self-organizing stable recognition categories in response to arbitrary sequences of input patterns (Figure 1). During training, the $ART_a$ module 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 ART modules are linked by an associative learning network and an internal controller that ensures autonomous system operation in real time. During test trials, the remaining patterns $\mathbf{a}^{(p)}$ are presented without $\mathbf{b}^{(p)}$, and their predictions at $ART_b$ are compared with $\mathbf{b}^{(p)}$.

Tested on a benchmark machine learning database in both on-line and off-line simulations, the ARTMAP system learns orders of magnitude more quickly, efficiently, and accurately than alternative algorithms, and achieves 100% accuracy after training on less than half the input patterns in the database. It achieves these properties by using an internal controller that






| | Predictive ART | Back Propagation |
|---|---|---|
| supervised | yes | yes |
| self-organizing | yes | no |
| real-time | yes | no |
| self-stabilizing | yes | no |
| learning: | fast or slow match | slow mismatch |

**Figure 1.** A Predictive ART, or ARTMAP, system includes two ART modules linked by an inter-ART associative memory. Internal control structures actively regulate learning and information flow. Back Propagation and Predictive ART both carry out supervised learning, but the two systems differ in many respects, as indicated.

conjointly maximizes predictive generalization and minimizes predictive error by linking predictive success to category size on a trial-by-trial basis, using only local operations. This computation increases the vigilance parameter $\rho_a$ of $ART_a$ by the minimal amount needed to correct a predictive error at $ART_b$. Parameter $\rho_a$ calibrates the minimum confidence that $ART_a$ must have in a category, or hypothesis, activated by an input $\mathbf{a}^{(p)}$ in order for $ART_a$ to accept that category, rather than search for a better one through an automatically controlled process of hypothesis testing. Parameter $\rho_a$ is compared with the degree of match between $\mathbf{a}^{(p)}$ and the top-down learned expectation, or prototype, that is read-out subsequent to activation of an $ART_a$ category. Search occurs if the degree of match is less than $\rho_a$. ARTMAP is hereby a type of self-organizing expert system that calibrates the selectivity of its hypotheses based upon predictive success. As a result, rare but important events can be quickly and sharply distinguished even if they are similar to frequent events with different consequences. Between input trials $\rho_a$ relaxes to a baseline vigilance $\overline{\rho_a}$. When $\overline{\rho_a}$ is large, the

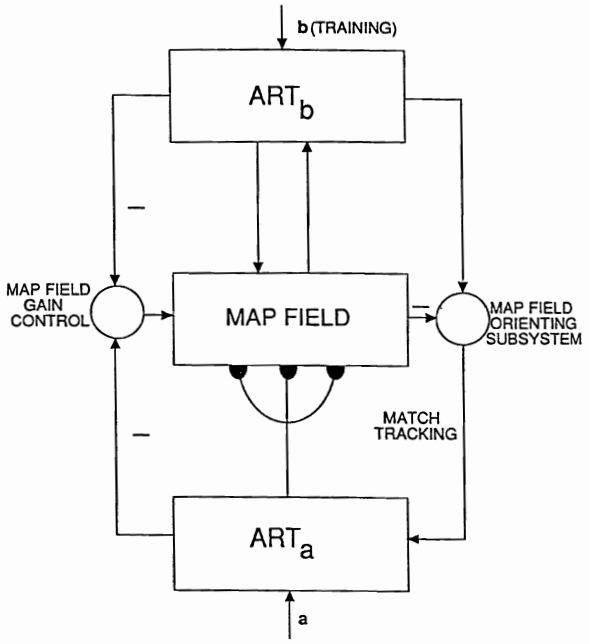



**Figure 2.** Block diagram of an ARTMAP system. Modules $ART_a$ and $ART_b$ self-organize categories for vector sets **a** and **b**. $ART_a$ and $ART_b$ are connected by an inter-ART module that consists of the Map Field and the control nodes called Map Field gain control and Map Field orienting subsystem. Inhibitory paths are denoted by a minus sign; other paths are excitatory.

system runs in a conservative mode, wherein predictions are made only if the system is confident of the outcome. Very few false-alarm errors then occur at any stage of learning, yet the system reaches asymptote with no loss of speed. Because ARTMAP learning is self-stabilizing, it can continue learning one or more databases, without degrading its corpus of memories, until its full memory capacity is utilized.

## II. The ARTMAP System

The main elements of an ARTMAP system are shown in Figure 2. Two modules, $ART_a$ and $ART_b$, read vector inputs **a** and **b**. If $ART_a$ and $ART_b$ were disconnected, each module would self-organize category groupings for the separate input sets. In the application described below, $ART_a$ and $ART_b$ are fast-learn ART 1 (Carpenter and Grossberg, 1987a) modules coding binary input vectors. $ART_a$ and $ART_b$ are here connected by an inter-ART module that in many ways resembles ART 1.



This inter-ART module includes a *Map Field* that controls the learning of an associative map from $ART_a$ recognition categories to $ART_b$ recognition categories. This map does not directly associate exemplars **a** and **b**, but rather associates the compressed and symbolic representations of families of exemplars **a** and **b**. The Map Field also controls match tracking of the $ART_a$ vigilance parameter. A mismatch at the Map Field between the $ART_a$ category activated by an input **a** and the $ART_b$ category activated by the input **b** increases $ART_a$ vigilance by the minimum amount needed for the system to search for and, if necessary, learn a new $ART_a$ category whose prediction matches the $ART_b$ category.

This inter-ART vigilance resetting signal is a form of "back propagation" of information, but one that differs from the back propagation that occurs in the Back Propagation network (Figure 1). For example, the search initiated by inter-ART reset can shift attention to a novel cluster of visual features that can be incorporated through learning into a new $ART_a$ recognition category. This process is analogous to learning a category for "green bananas" based on "taste" feedback. However, these events do not "back propagate" taste features into the visual representation of the bananas, as can occur using the Back Propagation network. Rather, match tracking reorganizes the way in which visual features are grouped, attended, learned, and recognized for purposes of predicting an expected taste.

The following sections describe ARTMAP simulations using a machine learning benchmark database. For a full specification of the ARTMAP system, and analysis of network dynamics, see Carpenter, Grossberg, and Reynolds (1991).

## III. ARTMAP Simulations: Distinguishing Edible and Poisonous Mushrooms

The ARTMAP system was tested on a benchmark machine learning database that partitions a set of vectors **a** into two classes. Each vector **a** characterizes observable features of a mushroom as a binary vector, and each mushroom is classified as edible or poisonous (Schlimmer, 1987a). The database represents the 11 species of genus *Agaricus* and the 12 species of the genus *Lepiota* described in **The Audubon Society Field Guide to North American Mushrooms** (Lincoff, 1981). These two genera constitute most of the mushrooms described in the **Field Guide** from the familiy *Agaricaceae* (order *Agaricales*, class *Hymenomycetes*, subdivision *Basidiomycetes*, division *Eumycota*). All the mushrooms represented in the database are similar to one another: "These mushrooms are placed in a single family on the basis of a correlation of characteristics that include microscopic and and chemical features..." (Lincoff, 1981, p.500).

The **Field Guide** warns that poisonous and edible species can be difficult to distinguish on the basis of their observable features. For example, the poisonous species *Agaricus californicus* is described as a "dead ringer" (p. 504) for the Meadow Mushroom, *Agaricus campestris*, that "may be known better and gathered more than any other wild mushroom in North America" (p. 505). This database thus provides a test of how ARTMAP and other machine learning systems distinguish rare but important events from frequently occurring collections of similar events that lead to different consequences.

The database of 8124 exemplars describes each of 22 observable features of a mushroom, along with its classification as poisonous (48.2%) or edible (51.8%). The 8124 "hypothetical examples" represent ranges of characteristics within each species; for example, both *Agaricus californicus* and *Agaricus campestris* are described as having a "white to brownish cap," so in the database each species has corresponding sets of exemplar vectors representing their range of cap colors. There are 126 different values of the 22 different observable features. A list of the observable features and their possible values is given in Table 1. For example, the observable feature of "cap-shape" has six possible values. Consequently, the vector inputs to $ART_a$ are 126-element binary vectors, each vector having 22 1's and 104 0's, to denote the values of an exemplar's 22 observable features. The $ART_b$ input vectors are (1,0) for poisonous exemplars and (0,1) for edible exemplars.

*A. Performance*

The ARTMAP system learned to classify test vectors rapidly and accurately, and system performance compares favorably with results of other machine learning algorithms applied to the same database. The STAGGER algorithm reached its maximum performance level of 95% accuracy after exposure to 1000 training inputs (Schlimmer, 1987b). The HILLARY algorithm achieved similar results (Iba, Wogulis, and Langley, 1988). The ARTMAP system consistently achieved over 99% accuracy with 1000 exemplars, even counting "I don't know" responses as errors. Accuracy of 95% was usually achieved with on-line training on 300–400 exemplars and with off-line training on 100–200 exemplars. In this sense, ARTMAP was an order of magnitude more efficient than the alternative systems. In addition, with continued training, ARTMAP predictive accuracy always improved to 100%. These results are elaborated below.

Almost every ARTMAP simulation was completed in under 2 minutes on an IRIS 4D computer, with total time ranging from about 1 minute for small training sets to 2 minutes for large training sets. This is comparable to 2–5 minutes on a SUN 4 computer. Each timed simulation included a

**TABLE 1: 22 Observable Features and their 126 Values**

| Number | Feature | Possible Values |
|---|---|---|
| 1 | cap-shape | bell, conical, convex, flat, knobbed, sunken |
| 2 | cap-surface | fibrous, grooves, scaly, smooth |
| 3 | cap-color | brown, buff, gray, green, pink, purple, red, white, yellow, cinnamon |
| 4 | bruises | bruises, no bruises |
| 5 | odor | none, almond, anise, creosote, fishy, foul, musty, pungent, spicy |
| 6 | gill-attachment | attached, descending, free, notched |
| 7 | gill-spacing | close, crowded, distant |
| 8 | gill-size | broad, narrow |
| 9 | gill-color | brown, buff, orange, gray, green, pink, purple, red, white, yellow, chocolate, black |
| 10 | stalk-shape | enlarging, tapering |
| 11 | stalk-root | bulbous, club, cup, equal, rhizomorphs, rooted, missing |
| 12 | stalk-surface-above-ring | fibrous, silky, scaly, smooth |
| 13 | stalk-surface-below-ring | fibrous, silky, scaly, smooth |
| 14 | stalk-color-above-ring | brown, buff, orange, gray, pink, red, white, yellow, cinnamon |
| 15 | stalk-color-below-ring | brown, buff, orange, gray, pink, red, white, yellow, cinnamon |
| 16 | veil-type | partial, universal |
| 17 | veil-color | brown, orange, white, yellow |
| 18 | ring-number | none, one, two |
| 19 | ring-type | none, cobwebby, evanescent, flaring, large, pendant, sheathing, zone |
| 20 | spore-print-color | brown, buff, orange, green, purple, white, yellow, chocolate, black |
| 21 | population | abundant, clustered, numerous, scattered, several, solitary |
| 22 | habitat | grasses, leaves, meadows, paths, urban, waste, woods |

**Table 1:** 126 values of 22 observable features represented in $ART_a$ input vectors.

total of 8124 training and test samples, run on a time-sharing system with non-optimized code. Each 1–2 minute computation included data read-in and read-out, training, testing, and calculation of multiple simulation indices.

### *B. On-Line Learning*

On-line learning imitates the conditions of a human or machine operating in a natural environment. An input **a** arrives, possibly leading to a prediction. If made, the prediction may or may not be confirmed. Learning ensues, depending on the accuracy of the prediction. Information about past inputs is available only through the present state of the system. Simulations of on-line learning by the ARTMAP system use each sample pair (**a**, **b**) as both a test item and a training item. Input **a** first makes a prediction that is compared with **b**. Learning follows as dictated by the internal rules of the ARTMAP architecture.

Four types of on-line simulations were carried out, using two different baseline settings of the $ART_a$ vigilance parameter $\rho_a$: $\overline{\rho_a} = 0$ (forced choice condition) and $\overline{\rho_a} = 0.7$ (conservative condition); and using sample replacement or no sample replacement. With sample replacement, any one of the 8124 input samples was selected at random for each input presentation. A given sample might thus be repeatedly encountered while others were still unused. With no sample replacement, a sample was removed from the input pool after it was first encountered. The replacement condition had the advantage that repeated encounters tended to boost predictive accuracy. The no-replacement condition had the advantage of having learned from a somewhat larger set of inputs at each point in the simulation. The replacement and no-replacement conditions had similar performance indices, all other things being equal. Each of the 4 conditions was run on 10 independent simulations. With $\overline{\rho_a} = 0$, the system made a prediction in response to every input. Setting $\overline{\rho_a} = 0.7$ increased the number of "I don't know" responses, increased the number of $ART_a$ categories, and decreased the rate of incorrect predictions to nearly 0%, even early in training. The $\overline{\rho_a} = 0.7$ condition generally outperformed the $\overline{\rho_a} = 0$ condition, even when incorrect predictions and "I don't know" responses were both counted as errors. The primary exception occurred very early in training, when a conservative system gives the large majority of its no-prediction responses.

Results are summarized in Table 2. Each entry gives the number of correct predictions over the previous 100 trials (input presentations), averaged over 10 simulations. For example, with $\overline{\rho_a} = 0$ in the no-replacement condition, the system made, on the average, 94.9 correct predictions and 5.1 incorrect predictions on trials 201–300. In all cases a 95% correct-prediction rate was achieved before trial 400. With $\overline{\rho_a} = 0$, a consistent



**TABLE 2: On-Line Learning**

| | Average number of correct predictions on previous 100 trials | | | |
|---|---|---|---|---|
| **Trial** | $\overline{\rho_a} = 0$ **no replace** | $\overline{\rho_a} = 0$ **replace** | $\overline{\rho_a} = 0.7$ **no replace** | $\overline{\rho_a} = 0.7$ **replace** |
| 100 | 82.9 | 81.9 | 66.4 | 67.3 |
| 200 | 89.8 | 89.6 | 87.8 | 87.4 |
| 300 | 94.9 | 92.6 | 94.1 | 93.2 |
| 400 | 95.7 | 95.9 | 96.8 | 95.8 |
| 500 | 97.8 | 97.1 | 97.5 | 97.8 |
| 600 | 98.4 | 98.2 | 98.1 | 98.2 |
| 700 | 97.7 | 97.9 | 98.1 | 99.0 |
| 800 | 98.1 | 97.7 | 99.0 | 99.0 |
| 900 | 98.3 | 98.6 | 99.2 | 99.0 |
| 1000 | 98.9 | 98.5 | 99.4 | 99.0 |
| 1100 | 98.7 | 98.9 | 99.2 | 99.7 |
| 1200 | 99.6 | 99.1 | 99.5 | 99.5 |
| 1300 | 99.3 | 98.8 | 99.8 | 99.8 |
| 1400 | 99.7 | 99.4 | 99.5 | 99.8 |
| 1500 | 99.5 | 99.0 | 99.7 | 99.6 |
| 1600 | 99.4 | 99.6 | 99.7 | 99.8 |
| 1700 | 98.9 | 99.3 | 99.8 | 99.8 |
| 1800 | 99.5 | 99.2 | 99.8 | 99.9 |
| 1900 | 99.8 | 99.9 | 99.9 | 99.9 |
| 2000 | 99.8 | 99.8 | 99.8 | 99.8 |

**Table 2:** On-line learning and performance in forced choice ($\overline{\rho_a} = 0$) or conservative ($\overline{\rho_a} = 0.7$) cases, with replacement or no replacement of samples after training.

correct-prediction rate of over 99% was achieved by trial 1400, while with $\overline{\rho_a} = 0.7$ the 99% consistent correct-prediction rate was achieved earlier, by trial 800. Each simulation was continued for 8100 trials. In all four cases, the minimum correct-prediction rate always exceeeded 99.5% by trial 1800 and always exceeded 99.8% by trial 2800. In all cases, across the total of 40 simulations summarized in Table 2, 100% correct prediction was achieved on the last 1300 trials of each run.

Note the relatively low correct-prediction rate for $\overline{\rho_a} = 0.7$ on the first 100 trials. In the conservative mode, a large number of inputs initially make no prediction. With $\overline{\rho_a} = 0.7$ an average total of only 2 *incorrect* predictions were made on each run of 8100 trials. Note too that Table 2 underestimates prediction accuracy at any given time, since performance almost always improves during the 100 trials over which errors are tabulated.

### C. Off-Line Learning

In off-line learning, a fixed training set is repeatedly presented to the system until 100% accuracy is achieved on that set. For training sets ranging in size from 1 to 4000 samples, 100% accuracy was almost always achieved after one or two presentations of each training set. System performance was then measured on the test set, which consisted of all 8124 samples not included in the training set. During testing no further learning occurred.

The role of repeated training set presentations was examined by comparing simulations that used the 100% training set accuracy criterion with simulations that used only a single presentation of each input during training. With only a few exceptions, performance was similar. In fact for $\overline{\rho_a} = 0.7$, and for small training sets with $\overline{\rho_a} = 0$, 100% training-set accuracy was achieved with single input presentations, so results were identical. Performance differences were greatest for $\overline{\rho_a} = 0$ simulations with mid-sized training sets (60–500 samples), when 2–3 training set presentations tended to add a few more $\mathrm{ART}_a$ learned category nodes. Thus, even a single presentation of training-then-testing inputs, carried out on-line, can be made to work almost as well as off-line training that uses repeated presentations of the training set. This is an important benefit of fast learning controlled by a match tracked search.

**1. Off-Line Forced-Choice Learning** The simulations summarized in Table 3 illustrate off-line learning with $\overline{\rho_a} = 0$. In this forced choice case, each $\mathrm{ART}_a$ input led to a prediction of poisonous or edible. The number of test set errors with small training sets was relatively large, due to the forced choice.



**TABLE 3: Off-Line Forced-Choice Learning**

| Training Set Size | Average % Correct (Test Set) | Average % Incorrect (Test Set) | Number of $\mathrm{ART}_a$ Categories |
|---|---|---|---|
| 3 | 65.8 | 34.2 | 1–3 |
| 5 | 73.1 | 26.9 | 1–5 |
| 15 | 81.6 | 18.4 | 2–4 |
| 30 | 87.6 | 12.4 | 4–6 |
| 60 | 89.4 | 10.6 | 4–10 |
| 125 | 95.6 | 4.4 | 5–14 |
| 250 | 97.8 | 2.2 | 8–14 |
| 500 | 98.4 | 1.6 | 9–22 |
| 1000 | 99.8 | 0.2 | 7–18 |
| 2000 | 99.96 | 0.04 | 10–16 |
| 4000 | 100 | 0 | 11–22 |

**Table 3:** Off-line forced choice ($\overline{\rho_a} = 0$) ARTMAP system performance after training on input sets ranging in size from 3 to 4000 exemplars. Each line shows average correct and incorrect test set predictions over 10 independent simulations, plus the range of learned $\mathrm{ART}_a$ category numbers.

Table 3 summarizes the average results over 10 simulations at each size training set. For example, with very small, 5-sample training sets, the system established between 1 and 5 $\mathrm{ART}_a$ categories, and averaged 73.1% correct responses on the remaining 8119 test patterns. Success rates ranged from chance (51.8%, 1 category) in one instance where all 5 training set exemplars happened to be edible, to surprisingly good (94.2%, 2 categories). The range of success rates for fast-learn training on very small training sets illustrates the statistical nature of the learning process. Intelligent sampling of the training set or, as here, good luck in the selection of representative samples, can dramatically alter early success rates. In addition, the evolution of internal category memory structure, represented by a set of $\mathrm{ART}_a$ category nodes and their top-down learned expectations, is influenced by the selection of early exemplars. Nevertheless, despite the individual nature of learning rates and internal representations, all the systems eventually converge to 100% accuracy on test set exemplars using only (approximately) 1/600 as many $\mathrm{ART}_a$ categories as there are inputs to classify.

**2. Off-Line Conservative Learning** As in the case of poisonous mushroom identification, it may be important for a system to be able to respond "I don't know" to a novel input, even if the total number of correct classifications thereby decreases early in learning. For higher values of the baseline vigilance $\overline{\rho_a}$, the ARTMAP system creates more $\mathrm{ART}_a$ categories during learning and becomes less able to generalize from prior experience than when $\overline{\rho_a}$ equals 0. During testing, a conservative coding system with $\overline{\rho_a} = 0.7$ makes no prediction in response to inputs that are too novel,

TABLE 4: Off-Line Conservative Learning

| Training Set Size | Average % Correct (Test Set) | Average % Incorrect (Test Set) | Average % No-Response (Test Set) | Number of $ART_a$ Categories |
|---|---|---|---|---|
| | 25.6 | 0.6 | 73.8 | 2–3 |
| | 41.1 | 0.4 | 58.5 | 3–5 |
| 5 | 57.6 | 1.1 | 41.3 | 8–10 |
| 0 | 62.3 | 0.9 | 36.8 | 14–18 |
| 0 | 78.5 | 0.8 | 20.8 | 21–27 |
| 25 | 83.1 | 0.7 | 16.1 | 33–37 |
| 50 | 92.7 | 0.3 | 7.0 | 42–51 |
| 00 | 97.7 | 0.1 | 2.1 | 48–64 |
| 000 | 99.4 | 0.04 | 0.5 | 53–66 |
| 000 | 100.0 | 0.00 | 0.05 | 54–69 |
| 000 | 100.0 | 0.00 | 0.02 | 61–73 |

Table 4: Off-line conservative ($\overline{\rho_a} = 0.7$) ARTMAP system performance after training on input sets ranging in size from 3 to 4000 exemplars. Each line shows average correct, incorrect, and no-response test set predictions over 10 independent simulations, plus the range of learned $ART_a$ category numbers.

and thus initially has a lower proportion of correct responses. However, the number of incorrect responses is always low with $\overline{\rho_a} = 0.7$, even with very few training samples, and the 99% correct-response rate is achieved for both forced choice ($\overline{\rho_a} = 0$) and conservative ($\overline{\rho_a} = 0.7$) systems with training sets smaller than 1000 exemplars.

Table 4 summarizes simulation results that repeat the conditions of Table 3 except that $\overline{\rho_a} = 0.7$. Here, a test input that does not make a 70% match with any learned expectation makes an "I don't know" prediction. Compared with the $\overline{\rho_a} = 0$ case of Table 3, Table 4 shows that larger training sets are required to achieve a correct prediction rate of over 95%. However, because of the option to make no prediction, the average test set error rate is almost always less than 1%, even when the training set is very small, and is less than .1% after only 500 training trials. Moreover, 100% accuracy is achieved using only (approximately) 1/130 as many $ART_a$ categories as there are inputs to classify.

### D. *Category Structure*

Each ARTMAP category code can be described as a set of $ART_a$ feature values on 1 to 22 observable features, chosen from 126 feature values, that are associated with the $ART_b$ identification as poisonous or edible. During learning, the number of feature values that characterize a given category is monotone decreasing, so that generalization within a given category tends to increase. The total number of classes can, however, also increase, which tends to decrease generalization. Increasing the number of training patterns hereby tends to increase the number of categories and



decrease the number of critical feature values of each established category. The balance between these opposing tendencies leads to the final net level of generalization.

Table 5 illustrates the long term memory structure underlying a 125-sample forced-choice simulation. Of the 9 categories established at the end of the training phase, 4 are identified as poisonous (P) and 5 are identified as edible (E). Each $ART_a$ category assigns a feature value to a subset of the 22 observable features. For example, Category 1 (poisonous) specifies values for 5 features, and leaves the remaining 17 features unspecified. The corresponding $ART_a$ weight vector has 5 ones and 121 zeros. Note that the features that characterize category 5 (poisonous) form a subset of the features that characterize category 6 (edible). Recall that this category structure gave 96.4% correct responses on the 7999 test set samples, which are partitioned as shown in the last line of Table 5. When 100% accuracy is achieved, a few categories with a small number of specified features typically code large clusters, while a few categories with many specified features code small clusters of rare samples.

Table 6 illustrates the statistical nature of the coding process, which leads to a variety of category structures when fast learning is used. Test set prediction accuracy of the simulation that generated Table 6 was similar to that of Table 5, and each simulation had a 125-sample training set. However, the simulation of Table 6 produced only 4 $ART_a$ categories, only one of which (category 1) has the same long term memory representation as category 2 in Table 5. Note that, at this stage of coding, certain features are uninformative. For example, no values are specified for features 1, 2, 3, or 22 in Table 5 or Table 6; and feature 16 (veil-type) always has the value "partial." However, performance is still only around 96%. As rare instances form small categories later in the coding process, some of these features may become critical in identifying exemplars of small categories.

## IV. Conclusion: Predictive ART

As we move freely through the world, we can attend to both familiar and novel objects, and can rapidly learn to recognize, test hypotheses about, and learn to name novel objects without unselectively disrupting our memories of familiar objects. This chapter has described some properties of a new self-organizing neural network architecture—called a Predictive ART or ARTMAP architecture—that is capable of fast, yet stable, on-line recognition learning, hypothesis testing, and adaptive naming in response to an arbitrary stream of input patterns.

The possibility of stable learning in response to an arbitrary stream of inputs is required by an autonomous learning agent that needs to cope with

TABLE 5

| # | Feature | 1=P | 2=E | 3=E | 4=E | 5=P | 6=E | 7=P | 8=P | 9=E |
|---|---|---|---|---|---|---|---|---|---|---|
| | cap-shape | | | | | | | | | |
| | cap-surface | | | | | | | | | |
| | cap-color | | | | | | | | | |
| | bruises? | | | | | | | yes | no | yes |
| | odor | | none | | | | none | | | |
| | gill-attachment | free | free | | free | free | free | free | free | free |
| | gill-spacing | close | | | close | close | close | close | close | close |
| | gill-size | | broad | | | | | | narrow | broad |
| | gill-color | | | | | | | | buff | |
| 0 | stalk-shape | | | | | | | | tapering | enlarged |
| 1 | stalk-root | | | | | | | | missing | club |
| 2 | stalk-surface-above-ring | | | smooth | smooth | smooth | smooth | smooth | smooth | smooth |
| 3 | stalk-surface-below-ring | | | smooth | | | | | | smooth |
| 4 | stalk-color-above-ring | | | | | white | white | white | pink | white |
| 5 | stalk-color-below-ring | | | | | | | white | | white |
| 6 | veil-type | partial | partial | partial | partial | partial | partial | partial | partial | partial |
| 7 | veil-color | white | white | | white | white | white | white | white | white |
| 8 | ring-number | one | | one | one | | one | one | one | one |
| 9 | ring-type | | | pendant | | | | | | |
| 0 | spore-print-color | | | | | | | | | |
| 1 | population | | | | | | | | | |
| 2 | habitat | | | | | | | | | |
| # coded/ category: | | 2367 | 1257 | 387 | 1889 | 756 | 373 | 292 | 427 | 251 |

Table 5: Critical feature values of the 9 category prototypes learned in the 125-sample simulation illustrated in Figure 4c ($\overline{\rho_a} = 0$). Categories 1, 5, 7 and 8 are identified as poisonous (P) and categories 2, 3, 4, 6, and 9 are identified as edible (E). These prototypes yield 96.4% accuracy on test set inputs.



TABLE 6

| # | Feature | 1=E | 2=P | 3=P | 4=E |
|---|---|---|---|---|---|
| 1 | cap-shape | | | | |
| 2 | cap-surface | | | | |
| 3 | cap-color | | | | |
| 4 | bruises? | | | no | |
| 5 | odor | none | | | |
| 6 | gill-attachment | free | free | | |
| 7 | gill-spacing | | | close | close |
| 8 | gill-size | broad | | | broad |
| 9 | gill-color | | | | |
| 10 | stalk-shape | | | | enlarging |
| 11 | stalk-root | | | | |
| 12 | stalk-surface-above-ring | | | | smooth |
| 13 | stalk-surface-below-ring | | | | |
| 14 | stalk-color-above-ring | | | | |
| 15 | stalk-color-below-ring | | white | | |
| 16 | veil-type | partial | partial | partial | partial |
| 17 | veil-color | white | white | white | |
| 18 | ring-number | | one | | one |
| 19 | ring-type | | | | pendant |
| 20 | spore-print-color | | | | |
| 21 | population | | | | |
| 22 | habitat | | | | |
| | # coded/category: | 3099 | 1820 | 2197 | 883 |

**Table 6:** Critical feature values of the 4 prototypes learned in a 125-sample simulation with a training set different from the one in Table 6. Prediction accuracy is similar (96.0%), but the $ART_a$ category boundaries are different.

unexpected events in an uncontrolled environment. One cannot *restrict* the agent's ability to process input sequences if one cannot *predict* the environment in which the agent must successfully function. The ability of humans to vividly remember exciting adventure movies is a familiar example of fast learning in an unfamiliar environment.

### *A. Fast Learning About 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 frequent events with very different consequences. Survival may hereby depend on fast learning in a *nonstationary* environment. Many learning schemes are, in contrast, slow learning models that average over individual event occurrences and are degraded by learning instabilities in a nonstationary environment (Carpenter, 1989; Carpenter and Grossberg, 1988; Grossberg, 1988).

### *B. Many-To-One and One-To-Many Learning*

An efficient recognition system needs to be capable of many-to-one

earning. For example, each of the different exemplars of the font for a pre-cribed letter may generate a single compressed representation that serves is a visual recognition category. This exemplar-to-category transforma-ion is a case of many-to-one learning. In addition, many different fonts—ncluding lower case and upper case printed fonts and scripts of various inds—can all lead to the same verbal name for the letter. This is a second ense in which learning may be many-to-one.

Learning may also be one-to-many, so that a single object can gener-te many different predictions or names. For example, upon looking at a anana, one may classify it as an oblong object, a fruit, a banana, a yellow anana, and so on. A flexible knowledge system may thus need to repre-ent in its memory many predictions for each object, and to make the best rediction for each different context in which the object is embedded.

### *. Control of Hypothesis Testing, Attention, and Learning by Predictive uccess*

Why does not an autonomous recognition system get trapped into arning only that interpretation of an object which is most salient given ie system's initial biases? One factor is the ability of that system to re-ganize its recognition, hypothesis testing, and naming operations based pon its predictive success or failure. For example, a person may learn visual recognition category based upon seeing bananas of various colors id associate that category with a certain taste. Due to the variability of olor features compared with those of visual form, this learned recognition itegory may incorporate form features more strongly than color features. owever, the color green may suddenly, and unexpectedly, become an im-ortant differential predictor of a banana's taste.

The different taste of a green banana triggers hypothesis testing that ifts the focus of visual attention to give greater weight, or salience, to ie banana's color features without negating the importance of the other atures that define a banana's form. A new visual recognition category n hereby form for green bananas, and this category can be used to accu-tely predict the different taste of green bananas. The new, finer category n form, moreover, without recoding either the previously learned generic presentation of bananas or their taste association.

Future representations may also form that incorporate new knowledge out bananas, without disrupting the representations that are used to edict their different tastes. In this way, predictive feedback provides one eans whereby one-to-many recognition and prediction codes can form rough time, by using hypothesis testing and attention shifts that support w recognition learning without forcing unselective forgetting of previous



knowledge.

### *D. Self-Organizing Expert System*

ARTMAP achieves its combination of desirable properties by acting as a type of self-organizing expert system. It incorporates the basic properties of all ART systems to carry out autonomous hypothesis testing and parallel memory search for appropriate recognition codes. Hypothesis testing terminates in a sustained state of resonance that persists as long as an input remains approximately constant. The resonance generates a focus of attention that selects the bundle of critical features common to the bottom-up input and the top-down expectation, or prototype, that is read-out by the resonating recognition category. Learning of the critical feature pattern occurs in this resonant and attentive state, hence the term *adaptive resonance.*

### *E. Conjointly Maximizing Generalization and Minimizing Predictive Error*

In summary, the ARTMAP system is designed to conjointly *maximize* generalization and *minimize* predictive error under *fast learning* conditions in *real time* in response to an *arbitrary ordering* of input patterns. Remarkably, the network can achieve 100% test set accuracy on a machine learning benchmark database, as described above. Each ARTMAP system learns to make accurate predictions quickly, in the sense of using relatively little computer time; efficiently, in the sense of using relatively few training trials; and flexibly, in the sense that its stable learning permits continuous new learning, on one or more databases, without eroding prior knowledge, until the full memory capacity of the network is exhausted.

## Acknowledgements

This research was supported in part by the Air Force Office of Scientific Research (AFOSR 90-0175 and AFOSR 90-0128), the Army Research Office (ARO DAAL-03-88-K0088), BP (98-A-1204), DARPA (AFOSR 90-0083), and the National Science Foundation (NSF IRI-90-00539). The authors wish to thank Cynthia E. Bradford and Carol Y. Jefferson for their valuable assistance in the preparation of the manuscript.

## References

1. Carpenter, G.A. (1989). Neural network models for pattern recognition and associative memory. *Neural Networks*, **2**, 243–257.

2. Carpenter, G.A. and Grossberg, S. (1987a). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, **37**, 54–115.
3. Carpenter, G.A. and Grossberg, S. (1987b). ART 2: Stable self-organization of pattern recognition codes for analog input patterns. *Applied Optics*, **26**, 4919–4930.
4. Carpenter, G.A. and Grossberg, S. (1988). The ART of adaptive pattern recognition by a self-organizing neural network. *Computer*, **21**, 77–88.
5. Carpenter, G.A., and Grossberg, S. (1990). ART 3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures. *Neural Networks*, **3**, 129–152.
6. 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. *IEEE Expert*, **6**.
7. Grossberg, S. (1988). Nonlinear neural networks: Principles, mechanisms, and architectures. *Neural Networks*, **1**, 17–61.
8. Iba, W., Wogulis, J., and Langley, P. (1988). Trading off simplicity and coverage in incremental concept learning. In **Proceedings of the 5th international conference on machine learning**. Ann Arbor, MI: Morgan Kaufmann, 73–79.
9. Lincoff, G.H. (1981). **The Audubon Society field guide to North American mushrooms**. New York: Alfred A. Knopf.
10. Schlimmer, J.S. (1987a). Mushroom database. UCI Repository of Machine Learning Databases. (aha@ics.uci.edu)
11. Schlimmer, J.S. (1987b). Concept acquisition through representational adjustment (Technical Report 87–19). Doctoral dissertation, Department of Information and Computer Science, University of California at Irvine.