Pergamon

Neural Networks, Vol. 9, No. 5, pp. 881–897, 1996

Printed in Great Britain
0893–6080/96 $15.00 + .00

0893–6080(95)00115–8

CONTRIBUTED ARTICLE

Gaussian ARTMAP: A Neural Network for Fast Incremental Learning of Noisy Multidimensional Maps

JAMES R. WILLIAMSON

Boston University

(*Received* 20 *February* 1995; *revised and accepted* 13 *September* 1995)

Abstract—*A new neural network architecture for incremental supervised learning of analog multidimensional maps is introduced. The architecture, called Gaussian ARTMAP, is a synthesis of a Gaussian classifier and an adaptive resonance theory (ART) neural network, achieved by defining the ART choice function as the discriminant function of a Gaussian classifier with separable distributions, and the ART match function as the same, but with the distributions normalized to a unit height. While Gaussian ARTMAP retains the attractive parallel computing and fast learning properties of fuzzy ARTMAP, it learns a more efficient internal representation of a mapping while being more resistant to noise than fuzzy ARTMAP on a number of benchmark databases. Several simulations are presented which demonstrate that Gaussian ARTMAP consistently obtains a better trade-off of classification rate to number of categories than fuzzy ARTMAP. Results on a vowel classification problem are also presented which demonstrate that Gaussian ARTMAP outperforms many other classifiers.*

Keywords—Pattern recognition, Adaptive resonance theory, ARTMAP, Incremental learning, Self-organization, Noisy data, Gaussian classifier, Radial basis function.

1. INTRODUCTION

Systems for incremental learning of multidimensional maps build and update an internal representation of the mapping on a case by case basis and typically without any a priori knowledge of the problem domain. For each new training sample, which consists of a pair of input and output vectors, this internal representation is refined in order to improve future prediction given a test sample, which consists solely of an input vector. Desirable characteristics of learning systems are as follows.

Parallel computation. Use simple local operations which are suitable for implementation in parallel hardware.

Fast learning. Learn the mapping quickly and reliably from as few training samples as possible.

Efficient representation. Minimize the storage requirement of the internal representation while maximizing predictive accuracy.

Resistant to noise. System's representation should remain efficient even if data are noisy. Training samples often contain incorrect or inconsistent input/output pairings, due to either errors in the collection data, or to the intrinsic discriminative insufficiency of the data features.

The development of incremental supervised learning systems has included a promising line of research investigating ARTMAP neural network architectures. The most prominent ARTMAP system for classifying analog data is fuzzy ARTMAP (FA), which has been shown to perform well in a number of benchmarks with respect to other learning systems (Carpenter et al., 1991a, 1992a, b). In this paper, a new ARTMAP system called Gaussian ARTMAP (GA) is introduced. GA satisfies the above criteria for incremental learning systems better than FA because it produces a more efficient representation and is more resistant to noise.

This paper is organized as follows. In Section 2, FA is briefly reviewed; two deficiencies of FA are

Acknowledgements: This research was supported in part by ARPA (ONR N00014-92-J-4015), the National Science Foundation (NSF IRI 90-00530), and the Office of Naval Research (ONR N00014-91-J-4100). The author wishes to thank Steve Grossberg and two anonymous reviewers for their valuable comments on the manuscript, and Natalya Markuzon for helpful discussions.

Requests for reprints should be sent to James R. Williamson, Department of Cognitive and Neural Systems, Boston University, 677 Beacon Street, Boston, MA 02215, USA; e-mail: jrw@cns.bu.edu

described and argued to stem from its fuzzy set category descriptions. In Section 3, an alternative network called Gaussian ARTMAP (GA) is proposed, which uses Gaussian defined categories by incorporating components of a Gaussian classifier into the ART choice and match functions. Section 4 describes the equations of Gaussian ART and Gaussian ARTMAP. In Section 5, GA and FA are evaluated on several data sets.

2. FUZZY ARTMAP

2.1. ART

The supervised learning ARTMAP architecture is an extension of the unsupervised clustering ART (adaptive resonance theory) architecture (Carpenter et al., 1991b). An ART network incrementally clusters its input into stable categories (Carpenter & Grossberg, 1987). The number of categories that are formed depends upon the vigilence parameter, ρ, which determines how "spread out" in feature space, according to the network's distance metric, samples coded by the same category may be. A vital ART concept is the separation of *choice* and *match* criteria. The choice function selects the network's current estimate of the category an input is most likely to belong to. The match function, on the other hand, determines if the chosen category's template is sufficiently similar to the input vector to satisfy ρ, the vigilance parameter. If the chosen category satisfies the match function, the system *resonates* and the category "learns": its template approaches the input vector. If not, the category is reset, and another category is chosen. If no existing category satisfies the match criterion, then a new category is recruited. Thus, ART incrementally produces the number of categories necessary to represent clusters of input samples, with the inclusivity of categories inversely related to ρ.

2.2. ARTMAP

ARTMAP extends ART into a supervised learning system by cleverly taking advantage of ART's unsupervised clustering mechanism (Carpenter et al., 1991b). In ARTMAP, the chosen ART categories (hidden units) learn predictions, which are mappings to output classes, during training. If a chosen ART category makes the wrong prediction, then the vigilance parameter ρ is temporarily raised to the level required to reset the category. This *match tracking* process guarantees that, for a given input sample, the category that resonates has a better match than all categories that are reset. Thus, the system organizes its clustering of the data based on predictive feedback from the labels it assigns to the clusters, as well as from how the data are distributed in feature space.

2.3. Fuzzy ARTMAP

The most prominent ARTMAP system for classifying analog data is fuzzy ARTMAP (Carpenter et al., 1992b). Fuzzy ART is an extension of the original binary ART 1 system to the analog domain through the use of the $\wedge$ AND fuzzy operator instead of the $\cap$ logical intersection (Carpenter et al., 1991b).

With FA, an input vector $I = I_1, \ldots, I_M$ is complement coded into $I := I, I^c = \{I_1, \ldots, I_M, I_{M+1} = 1 - I_1, \ldots, I_{2M} = 1 - IM\}$. Each category j is initialized with a weight vector $\mathbf{w}_j = \mathbf{w}_1 = \cdots = \mathbf{w}_{2M} = 1$. The choice function,

$$J = \arg\max_j \left(T_j(I) = \frac{|I \wedge \mathbf{w}_j|}{\alpha + |\mathbf{w}_j|} \right), \tag{1}$$

picks the non-reset category J with a combination of the best matching weight vector $|I \wedge \mathbf{w}_j|$ and the smallest (i.e., most specific) weight vector $|\mathbf{w}_j|$. The relative contributions of these components is determined by α, the choice parameter. If α is small, categories with small weight vectors $|\mathbf{w}_j|$, and thus large categories in feature space, are favored; if α is large, the opposite.

The match criterion,

$$\frac{|I \wedge w_J|}{|I|} \geqslant \rho, \tag{2}$$

requires that a chosen category's weight vector be sufficiently close to the input vector. Due to complement coding, the denominator $|I|$ in (2) is constant and can be ignored. Fast learning simply updates the weight vector of the chosen category with

$$\mathbf{w}_J := I \wedge \mathbf{w}_J. \tag{3}$$

Thus, the length of all weight vectors is non-increasing over time.

Each category's $2M$-dimensional weight vector can be viewed as an M-dimensional hyperrectangle, where the minimum and maximum values of the hyperrectangle in each dimension correspond to the minimum and maximum values of all the samples coded by that category if fast learning is used. Thus, a small α gives advantage to large inclusive categories (large hyperrectangles). This generally causes the system to create fewer categories than it would if α were large.

2.4. Fuzzy ARTMAP Deficiencies

Fuzzy ARTMAP has been shown to perform well in certain benchmarks with respect to other learning systems. However, two potential weaknesses of FA may be noted: (1) sensitivity to noise, and (2) inefficiency of fuzzy categories.

2.4.1. *Sensitivity to Noise.* When a FA category j makes a false prediction during training, it is reset and another category j' is chosen. If $|\mathbf{w}_{j'} \wedge I| = |\mathbf{w}_{j'}|$ and $|\mathbf{w}_j \wedge I| = |\mathbf{w}_j|$ (i.e., the training sample falls within the intersection of the two hyperrectangles), then $|\mathbf{w}_{j'}| > |\mathbf{w}_j|$, and thus category j' must have a smaller hyperrectangle than category j. When training data are noisy, so that regions of feature space essentially map randomly to different predictions, FA proliferates categories. Because satisfaction of each reset often requires a smaller, more specific category as outlined above, a succession of contradicting predictions in nearby random locations results in the continual recruitment of new categories. This category proliferation problem is partly due to the fact that the choice and match functions are flat within a category's hyperrectangle, and partly due to the use of fast learning. The problem of category proliferation in noise has been addressed by restricting the invocation of match tracking through an appropriate use of slow learning of class predictions that enables the network to learn conditional probabilities in a nonparametric setting (Carpenter et al., 1994). Here it is shown how category proliferation can be limited even during fast learning by modifying network dynamics.

2.4.2. *Inefficiency of Fuzzy Categories.* A related problem with FA is the potential inefficiency of fuzzy categories for representing distributions of data. Each category is represented by perhaps the simplest statistics about its data: the minimum and maximum values in each dimension, which are learned to conjointly minimize predictive error and maximize predictive generalization. A hyperrectangle represents the range of acceptable category vectors. Such a representation is perhaps best suited to data that are uniformly distributed within hyperrectangles.

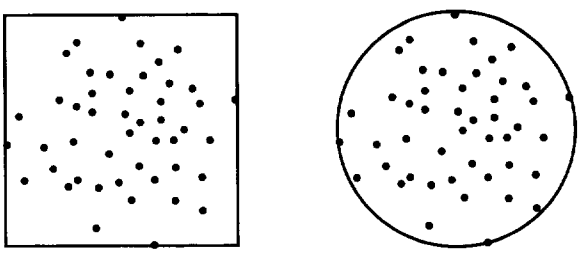

FIGURE 1. Two ways to fit a cluster of data: with a square and a circle.

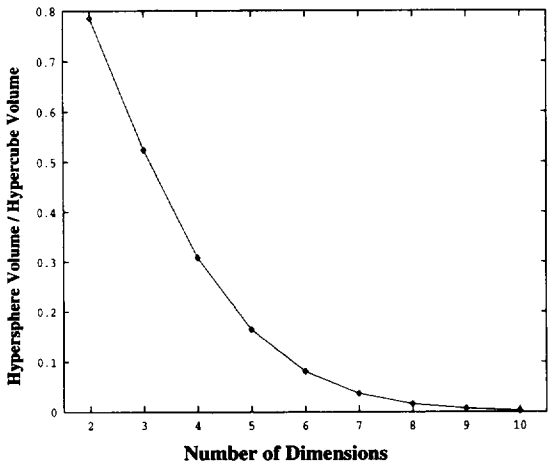

FIGURE 2. Ratio of volume of hypersphere to volume of hypercube with same diameter, as a function of the number of dimensions.

A "typical" cluster of real data may, however, be better characterized by the data points shown in Figure 1. The cluster of points is bounded by a fuzzy category forming a square in two dimensions (left), but is better fit by a bounding circle (right). Note that in the corners of the square, the category has inferred the existence of data where no evidence exists. If this inference turns out to be nonpredictive, then new categories may need to be created to "chip away at" the corners, in order to provide correct classification in those areas of feature space.

Figure 2 illustrates the way that this problem scales to higher dimensions by plotting a ratio, the volume of a hypershere divided by that of a hypercube with equal diameter, as a function of dimension. As feature space approaches 10 dimensions, the volume of a fuzzy category is dominated by the corners, for which little or no evidence may exist.

3. GAUSSIAN ARTMAP

To deal more efficiently with problems of category proliferation in noise and category shape, a new ART module called Gaussian ART is introduced, which uses categories defined as Gaussian distributions. Gaussian ART is incorporated into an ARTMAP architecture to create Gaussian ARTMAP (GA). The motivations for using Gaussian distributions are that: (1) Gaussian-defined choice and match functions, which monotonically increase toward a category's center, should produce less proliferation of categories in noise than do FA's flat choice and match functions; and (2) Gaussian distributions have useful generalization properties in high dimensional spaces (Duda & Hart, 1973; Powell, 1987; Broomhead & Lowe, 1988; Poggio & Girosi, 1989).

Components of a Gaussian classifier are incorporated into an ART module to create Gaussian ART.

This network has the familiar properties of ART networks because categories are incrementally formed to represent clusters of input samples, and the inclusivity of the categories is inversely related to a vigilance parameter, ρ. The novelty of Gaussian ART is that each ART category is defined as a Gaussian distribution, with a mean and standard variation in each dimension, and an a priori probability.

The choice function picks the most likely category for a given input. A category's likelihood is determined by the likelihood that the input belongs to its distribution, as well as by the category's a priori probability. The match function, on the other hand, is based on how well the input fits the category's distribution, which is normalized to a unit height.

When Gaussian ART is extended into Gaussian ARTMAP, the prediction of an output class during testing is interpreted as picking the class with the highest net probability. Therefore, all category predictions are summed to yield the most likely net prediction of a class, rather than basing the prediction on the maximum ART category, as in FA (but see also Carpenter and Ross, 1995).

Gaussian ARTMAP is essentially an incremental learning Gaussian classifier in which each output class is determined during training to correspond to any number of sources of Gaussianly distributed data. One limitation in this analogy is that GA

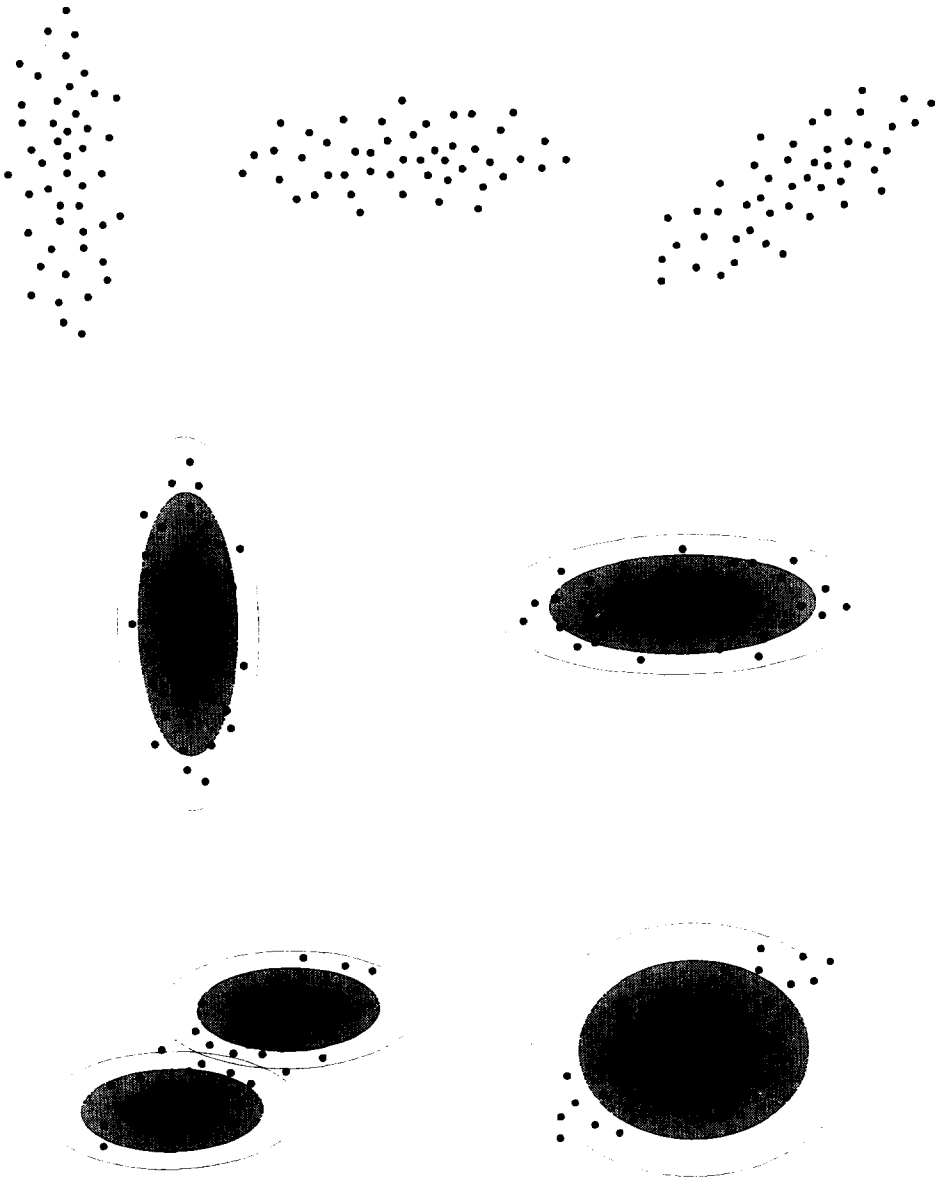

FIGURE 3. Top: Data which vary independently in each dimension (left, middle), and which covary between dimensions (right). Middle: GA categories, which are defined by separable Gaussian distributions, can capture independent variance well. Shown are Gaussian distributions that fit the independently varying data. Bottom: GA categories cannot capture covarying data as well. Two possibilities for fitting these data are two smaller distributions (left), and one larger distribution (right).

can only define its categories with separable Gaussian distributions. This limitation is necessary so that GA uses only simple operations that can be implemented in parallel. Figure 3 (top) illustrates three possible ways that data can be distributed in two dimensions. A GA category can easily fit data that vary independently in each dimension (middle), but cannot easily fit data that covary between dimensions (bottom). If GA were to represent covariance, then each category would need to store a covariance matrix, and classification would require computing the determinant and inverse of this matrix. By using only separable Gaussians, on the other hand, GA has storage and computational requirements similar to those of FA. One implication of this limitation, which also holds true for FA, is that GA most efficiently represents data that are uncorrelated across dimensions. It is interesting to note that humans also seem to have trouble representing covariance between dimensions (Kruschke, 1992).

4. GAUSSIAN ARTMAP EQUATIONS

4.1. Gaussian ART

4.1.1. *Categories.* Each Gaussian ART category j is defined by an M-dimensional vector μ_j representing its mean, σ_j representing its standard deviation, and a scalar n_j representing its count, the number of training samples it has coded. Thus, each Gaussian ART category requires $2M+1$ components to represent an M-dimensional input, $I = (I_1, \ldots, I_M)$.

4.1.2. *Category Choice.* During training, the category whose Gaussian distribution is the most probable "source" for input I is chosen. The a posteriori probability of category j given input I is

$$P(j|I) = \frac{p(I|j)P(j)}{p(I)}. \tag{4}$$

Categories are defined by separable Gaussian distributions, so the conditional density of I given category j is

$$p(I|j) = \frac{1}{(2\pi)^{M/2}\prod_{i=1}^{M}\sigma_{ji}} \exp\left(-\frac{1}{2}\sum_{i=1}^{M}\left(\frac{\mu_{ji}-I_i}{\sigma_{ji}}\right)^2\right), \tag{5}$$

and the a priori probability of j is simply

$$P(j) = \frac{n_j}{\sum_{j'=1}^{N} n_{j'}}, \tag{6}$$

where N is the number of categories. The density $p(I)$ in (4) is ignored because it is the same for all categories. For computational ease, a discriminant function $g_j()$ is used to evaluate each category, obtained by taking the log of the numerator in (4) with the dimensional scaling factor, $(2\pi)^{M/2}$, discounted (see Duda & Hart, pp. 22–31),

$$\begin{aligned} g_j(I) &= \log((2\pi)^{M/2}p(I|j)P(j)) \\ &= -\frac{1}{2}\sum_{i=1}^{M}\left(\frac{\mu_{ji}-I_i}{\sigma_{ji}}\right)^2 - \log\left(\prod_{i=1}^{M}\sigma_{ji}\right) + \log(P(j)). \end{aligned} \tag{7}$$

The non-reset ART category J with maximum discriminant function is chosen,

$$J = \arg\max_j(g_j(I)). \tag{8}$$

4.1.3. *Category Resonance and Reset.* If a chosen category's *match* value does not satisfy the ART vigilance parameter, ρ, then the category is reset. Category match is determined by how well input I matches with the shape of category J's distribution, which is normalized to a unit height,

$$\begin{aligned} g'_J(I) &= \log\left((2\pi)^{M/2}\left(\prod_{i=1}^{M}\sigma_{ji}\right)p(I|j)\right) \\ &= -\frac{1}{2}\sum_{i=1}^{M}\left(\frac{\mu_{ji}-I_i}{\sigma_{ji}}\right)^2 \\ &= g_J(I) - \log(P(J)) + \log\left(\prod_{i=1}^{M}\sigma_{ji}\right). \end{aligned} \tag{9}$$

If $g'_J(I) > \rho$, then the category resonates; otherwise it is reset. Once a category is reset, it remains inactive until presentation of the next input. If no committed ART category meets the vigilance condition, then an uncommitted category J', with $n_{J'} = 0$, is chosen.

Figure 4 shows a 1-D example of the match and choice functions, given three categories, A, B, and C, and an input sample, I. In this example, the first category chosen is A (top), however A does not meet the match criterion (bottom), so it is reset. The next category chosen is C, which meets the match criterion, and thus resonates. Therefore, category C learns input I: C's mean and standard deviation are updated by I, and its count is incremented.

4.1.4. *Learning.* When category J learns an input sample I, its count, mean, and standard deviation variables are updated to represent the sample count, mean, and standard deviation,

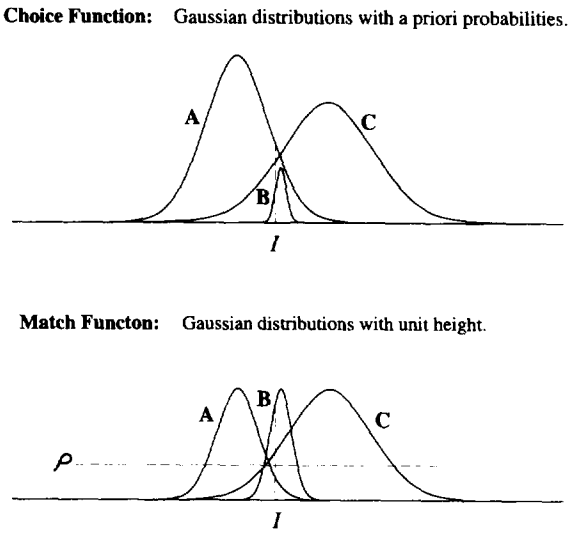

FIGURE 4. Example in one dimension of choice and match functions of three different GA categories. Input sample is denoted by *I*. In this example, category *A* wins the choice competition. However, category *A*'s distribution, normalized to unit height, is insufficient to meet the match criterion, determined by ρ. Therefore, category *A* is reset, and the category with the next highest choice function, category *C*, is chosen. Category *C* meets the match criterion, so category *C* resonates, and learns the input: its mean, standard deviation, and count variables are updated.

$$n_J := n_J + 1, \tag{10}$$

$$\mu_J := (1 - n_J^{-1})\mu_J + n_J^{-1} I, \tag{11}$$

$$\sigma_{Ji} := \begin{cases} \sqrt{(1 - n_J^{-1})\sigma_{Ji}^2 + n_J^{-1}(\mu_{Ji} - I_i)^2} & \text{if} \quad n_J > 1, \\ \gamma \quad \text{otherwise.} \end{cases} \tag{12}$$

The initial standard deviation, γ, determines the isotropic spread in feature space of a new category's distribution about its first sample.

4.1.5. *Input Normalization.* Gaussian ARTMAP can use inputs of any value. Because categories are initialized with a constant standard deviation γ in each dimension, however, it is usually desirable that inputs have roughly equal standard deviations in each dimension.

4.2. Gaussian ARTMAP

The Gaussian ART module plays the same role within the ARTMAP architecture as does an ART 1 module (Carpenter et al., 1991a), or a fuzzy ART module (Carpenter et al., 1992b). The most basic ARTMAP system is presented here. For a full network description of ARTMAP, see Carpenter et al. (1991a, b, 1992a, b).

4.2.1. *Training.* When an ART category J is chosen for the first time during training, it is assigned the prediction, K, of the current training sample,

$$\Omega(J) = K. \tag{13}$$

The function $\Omega(\,)$ maps category J to its prediction, class K. This function is generally many-to-one, so that $J \in \Omega^{-1}(K)$. If category J is again chosen in response to another training sample, and its prediction K' is incorrect ($K' \neq K$), then *match tracking* is invoked. The vigilance parameter is raised to the value of the category's match function,

$$\rho = g_J'(I), \tag{14}$$

and category J is reset. Match tracking assures that a correct prediction comes from a category whose distribution is a better match to the training sample than all reset categories. Upon presentation of the next training sample, ρ is reassigned its baseline value, $\rho = \bar{\rho}$.

4.2.2. *Testing.* During training, each prediction is made by the category with the maximum discriminant function in (8), and therefore is the maximum estimated likelihood of being the source for that prediction. During testing, on the other hand, the goal is to make the best prediction, not to assign credit to the most deserving category. Therefore, the prediction with the maximum estimated probability is chosen, where the probability estimates are obtained by summing the activations of all the categories that map to each prediction. The prediction K' with the maximum net probability is chosen,

$$K' = \arg\max_k \left(\sum_{j \in \Omega^{-1}(k)} \exp(g_j(I)) \right). \tag{15}$$

4.2.3. *Voting.* FA is sensitive to the order of its training samples. By independently training different FA systems on different orderings of the same data, and combining their predictions during testing by voting, performance can be significantly improved (Carpenter et al., 1992b). GA also benefits from combining the outputs of independently trained systems, although the benefit typically seems to be smaller than for FA, presumably because GA is less sensitive to the order of training samples. Just as a single GA prediction is based on net probabilities, determined by the sum of all category predictions in (15), so is GA voting also based on the sum of all category predictions across different GA systems,

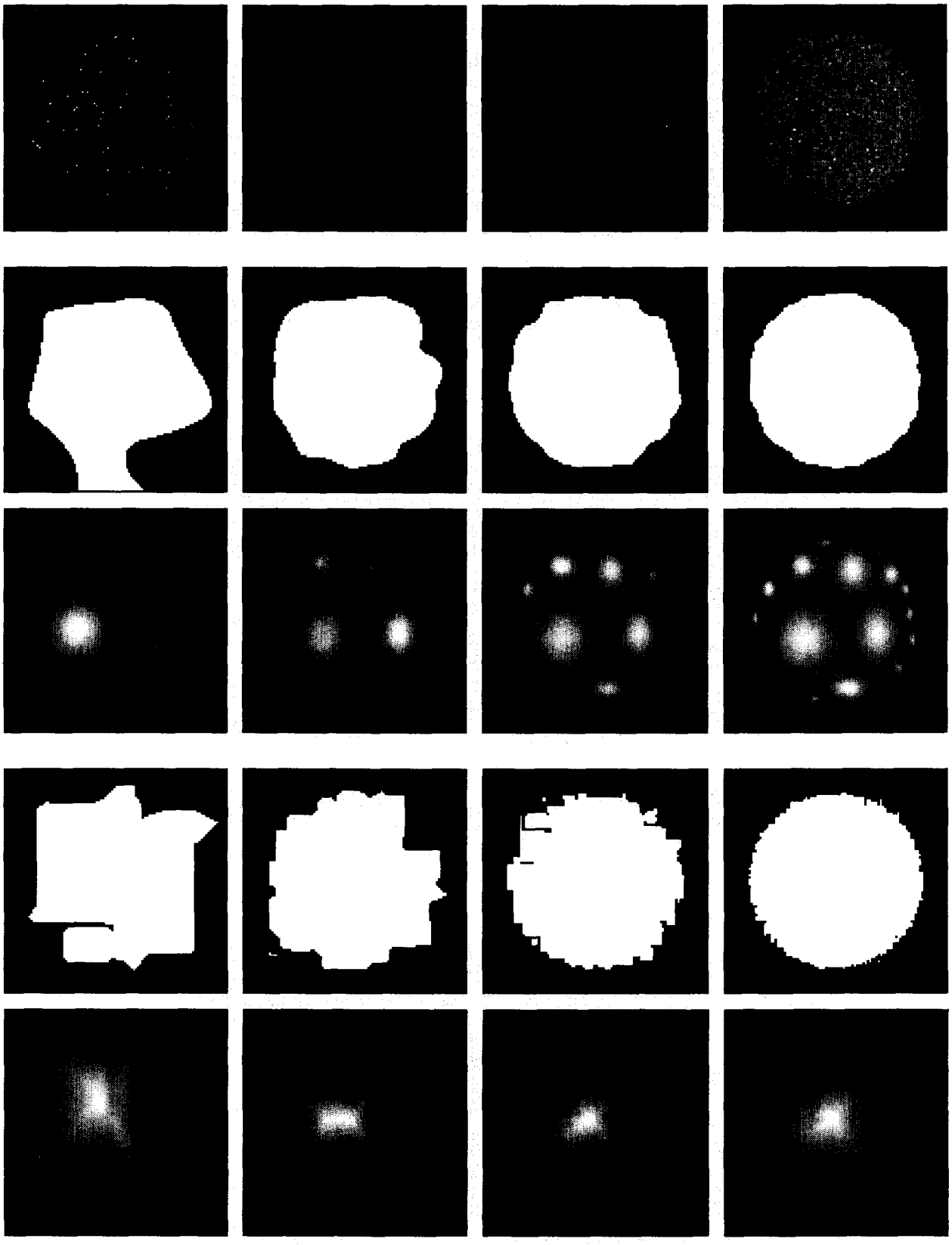

FIGURE 5. Circle-in-square. 1st Row: From left to right, 10 to the power 2, 3, 4, and 5 training samples. 2nd Row: GA (with $\gamma = 0.5$) decision regions. 3rd Row: Underlying GA difference of discriminant functions. 4th Row: FA (with $\alpha = 0.1$) decision regions. 5th Row: Underlying FA difference in discriminant functions.

$$K' = \arg\max_k \left(\sum_{v=1}^{V} \sum_{j \in \Omega_v^{-1}(k)} \exp(g_{v,j}(I)) \right), \qquad (16)$$

where V is the number of GA systems which are voting. While the operation in (16) would more accurately be called "summing" than "voting", I retain the latter term for historical consistency with FA.

4.2.4. *Comparison with Potential Functions and Gaussian Classifier.* It is interesting to note that if training results in each category coding only one sample, then GA becomes identical to the method of potential functions using Gaussian distributions. If training results in each output class corresponding to only one ART category, on the other hand, then GA becomes identical to a Gaussian classifier with separable Gaussians. Therefore, GA's extremes of minimal and maximal code compression correspond respectively to classification with potential functions and with a Gaussian classifier.

5. SIMULATIONS

Both FA and GA have internal parameters which affect the number of categories created during training. One of these is the vigilance parameter, ρ. Since this parameter has the same function in both systems, its baseline value is set to zero for both systems in all simulations. Each system has another parameter affecting the number of categories created, which has no analog in the other system. In FA this is α, the choice parameter. As mentioned earlier, α covaries with the number of categories created by FA. In GA, this parameter is γ, the initial standard deviation of categories, which inversely covaries with the number of categories created. FA and GA can thus be evaluated under conservative and non-conservative regimes by varying α and γ.

5.1. Circle-in-Square

The circle-in-the-square problem requires identification of which points lie inside and which outside a circle lying within a square of twice its area (Carpenter et al., 1991b). The performances of FA and GA were evaluated based on the number of categories created and on test error rate. Figure 5 (top row) shows the training data with, from left to right, 10 to the power 2, 3, 4, and 5 samples. Training samples belonging to the circle are shown in white, and those belonging to the background are shown in black. Testing consisted of 10,000 samples evenly spaced across the image. FA was evaluated with $\alpha = 0.1$ and $\alpha = 1.0$, and GA with $\gamma = 1.0$ and $\gamma = 0.5$.

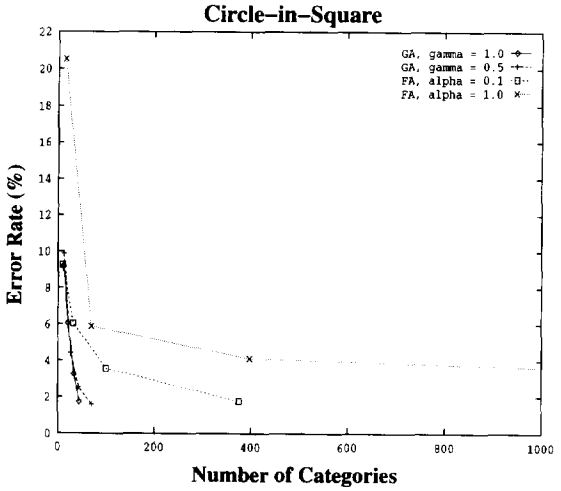

FIGURE 6. Circle-in-square test results. Graph jointly plots error rate (abscissa) and number of categories (ordinate). Along each curve (from left to right), each successive point corresponds to training on a larger training set. So, each curve shows the error rate and number of categories resulting from training on 10 to the power 2, 3, 4, and 5 training samples.

Figure 6 jointly plots the number of categories (ordinate) and error rate (abscissa) for the different number of training samples, with successive points along each line corresponding to training with 100, 1000, 10,000, and 100,000 training samples. The lower left of this graph is the desirable zone: low error rate with few categories. GA, with either value of γ, outperforms FA, with either value of α, because it achieves an equally low error rate using far fewer categories.

Figure 5 (second and third rows) visually illustrates GA's classification results, with $\gamma = 0.5$, corresponding to the training samples above. In the second row is shown the decision regions, and in the third row the difference of the sum of discriminant functions in (8) for the two classes,

$$\sum_{j \in \Omega^{-1}(1)} \exp(g_j(I)) - \sum_{j \in \Omega^{-1}(0)} \exp(g_j(I)). \qquad (17)$$

Note that large Gaussian "blobs" appear at the beginning of training (left), and as training progresses these blobs are tightened, and smaller blobs (additional categories) are added to refine the decision boundaries.

Figure 5 (fourth and fifth rows) shows the corresponding results for FA with $\alpha = 0.1$. Note that the decision boundaries are more choppy than those for GA, and that the difference in the maximum discriminant functions for the two classes,

$$\max_{j \in \Omega^{-1}(1)} T_j(I) - \max_{j \in \Omega^{-1}(0)} T_j(I), \qquad (18)$$

is not as revealing as the corresponding GA difference in (17).

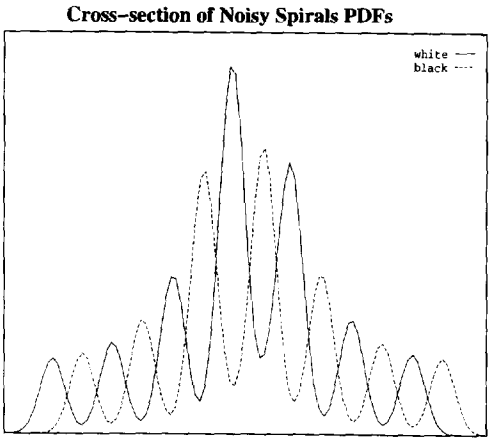

FIGURE 7. Cross-section, vertically down the middle, of nested spirals probability density functions (PDFs) using Gaussianly distributed noise.

5.2. Noisy Nested Spirals

5.2.1. *Gaussianly Distributed Noise.* Now that we have seen how the two systems perform on the relatively simple circle-in-square problem, it is interesting to see how they do on a more difficult problem that contains both noise and multimodally distributed data. One such task is that of discriminating noisy nested spirals (Carpenter et al., 1995). In this task, a square of range [0..1] in each dimension contains two intertwined spirals that a classifier attempts to discriminate. In our implementation of this problem, each spiral consisted of 97 isotropic Gaussian distributions centered along the spiral, each Gaussian having a standard deviation of 0.025. Figure 7 illustrates the large amount of overlap between the two classes, showing a vertical cross-section of the PDFs taken in the middle of the image. The task differs from the one in Carpenter et al. (1995) in that training samples were drawn randomly from all 194 Gaussians, rather than being restricted to 20 samples from each Gaussian, and the Gaussians were defined with a larger standard deviation (0.025 rather than 0.01).

For comparison with the circle-in-square results shown in Figure 5, GA and FA were evaluated with the same parameters, $\gamma = 0.5$, and $\alpha = 0.1$, respectively. In addition, FA was evaluated with two methods introduced in Carpenter et al. (1995) for coping with noisy data. These methods combine restriction of match tracking with slow learning of the mappings from chosen categories to output classes in order to obtain nonparametric probability estimations. The two methods differ in how match tracking is restricted. In the *slow learning* method, match tracking can only take place after enough evidence of a wrong prediction from a given category has been accumulated via the slow learning process. In the *max nodes* method, match tracking is completely turned off after a certain number of categories are created. Because GA is an ARTMAP system, it can also use the slow learning and max nodes methods to improve its resistance to noise, however here the performance of GA with only fast learning is compared to that of FA with the additional noise resistance mechanisms.

In Carpenter et al. (1995), nine different systems were independently trained on different orderings of the training data, and their outputs averaged. Here, the output of a single trained system was evaluated, rather than the outputs of nine different systems, so more nodes (500 rather than 75) were used in the max nodes method than in Carpenter et al. (1995). Otherwise, the slow learning and max nodes parameters were the same as in Carpenter et al. (1995).

Figure 8 (top row) shows the training data with, from left to right, 10 to the power 2, 3, 4, and 5 samples. Figure 8 (second and third rows) illustrates GA's classification results. After 10,000 training samples, GA has captured the form of the spirals quite well; its decision regions are further refined following 100,000 training samples. Figure 9 illustrates FA's classification results. The top row shows the fast learning results, the middle row the slow learning results, and the bottom row the max nodes results. With fast learning, FA creates very many categories, and the resulting decision regions visually resemble the noise. With slow learning, fewer categories are created, and the results are improved, yet still quite noisy. With max nodes, the decision regions markedly improve after the number of categories is restricted (far right), yet the results are not as good as GA's results in Figure 8.

Figure 10 plots the performance of GA and FA on 100,000 randomly generated test samples. GA converges to near the optimal error rate of 20% with fewer than 800 categories. With fast learning and slow learning, FA creates far more categories without significantly reducing its error rate. With max nodes, FA's error rate begins to approach that of GA after the number of categories is bounded at 500.

5.2.2. *Uniformly Distributed Noise.* A shortcoming of the above comparisons between FA and GA is that GA may have an unfair advantage due to the nature of the problems. The circularly shaped decision boundaries are more easily fit by Gaussian distributions, as are the Gaussian noise distributions in the noisy nested spirals problem. In order to remove this latter possible confound, training and testing on the same noisy nested spirals problem was repeated, except with samples perturbed with uniformly rather than Gaussianly distributed noise. The uniformly distributed noise had the same standard deviation (0.025) as the previous Gaussianly distributed noise.

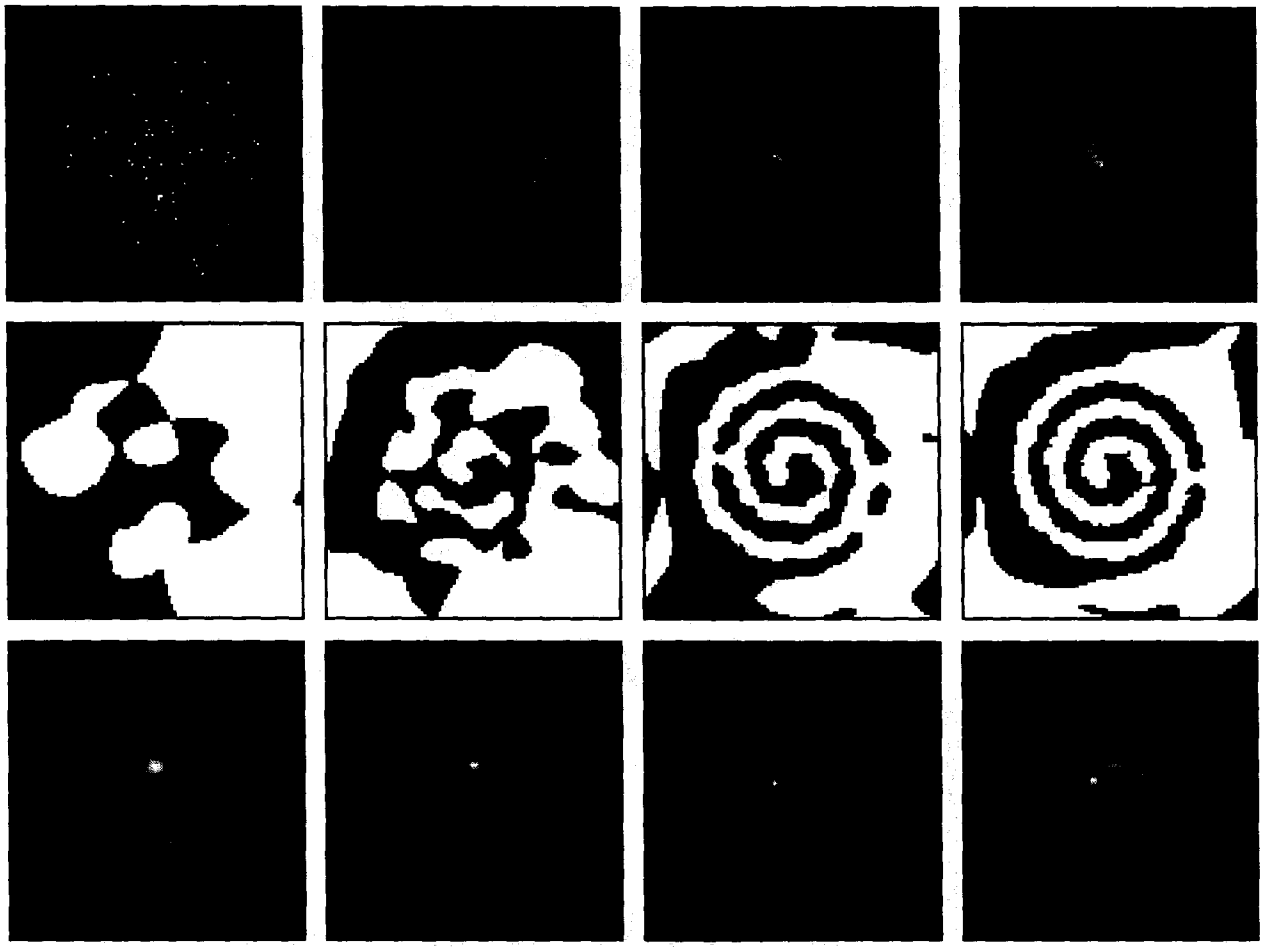

FIGURE 8. Nested spirals with Gaussianly distributed noise. 1st Row: From left to right, 10 to the power 2, 3, 4, and 5 training samples. 2nd Row: GA (with $\gamma = 0.5$) decision regions. 3rd Row: Underlying GA difference in discriminant functions.

Figure 11 plots the results on uniformly distributed noise, which correspond to the results on Gaussianly distributed noise plotted in Figure 10. The high similarity in the two sets of results indicates that GA's advantage over FA depends much more on the amount of noise rather than on how the noise is distributed.

5.3. Letter Image Recognition

Frey and Slate (1991) developed a benchmark letter image recognition task. In this task, data consist of 16 features obtained from machine generated images of alphabetical characters (A to Z). The classification problem is to predict the correct letter from the 16 features. Classification difficulty stems from the fact that the characters are generated from 20 different fonts, are randomly warped, and only simple features such as the total number of "on" pixels, and the size and position of a box around the "on" pixels, are used. The database consists of 20,000 samples, the first 16,000 of which are used for training, and the last 4000 for testing. The database is archived in the UCI Repository of Machine Learning Databases and Domain Theories, maintained by D. Aha and P. Murphy (ml_repository@ics.uci.edu).

Frey and Slate (1991) tested several variations of Holland-style genetic classifiers, and achieved maximum performance of a little over 80%. Carpenter et al. (1992a) obtained dramatically better results with FA, with a maximum performance of 96% correct. This result was obtained using a subset of 11 of the 16 features. The first five features apparently have little discriminative value, and FA achieves better classification results without them.

FA and GA were evaluated with all 16 features, as well as with the reduced set of 11 features. To compare FA and GA, each parameter variation of each system was evaluated on the data by independently training the classifier five times, each for a total of 20 epochs (iterations through the data), or until the network equilibrated. The average performance, as well as the performance with voting, was obtained. Between each training epoch, the order of samples was randomly scrambled. FA was evaluated with the same parameters used in Carpenter et al. (1992a), $\alpha = 1.0$ and $\alpha = 0.1$. The input data were renormalized to have a range of [0..1] in each dimension for

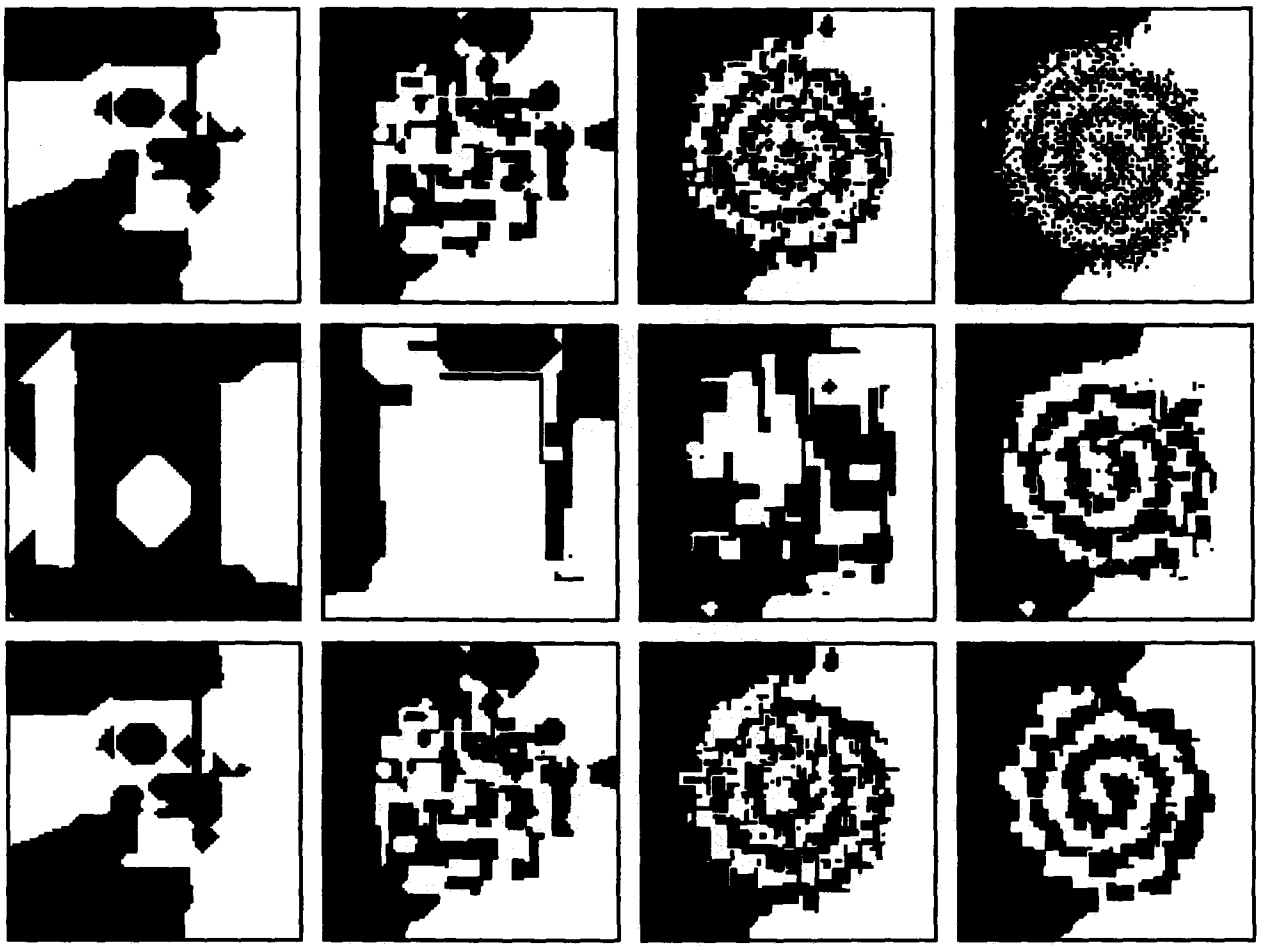

FIGURE 9. Nested spirals with Gaussianly distributed noise: FA results with $\alpha = 0.1$. 1st Row: FA decision regions with fast learning. 2nd Row: FA decision regions with slow learning. 3rd Row: FA decision regions with max nodes.

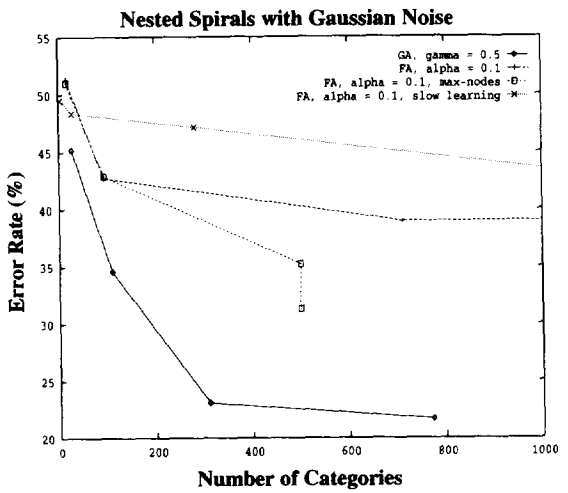

FIGURE 10. Nested spirals with Gaussianly distributed noise: test results. Graph jointly plots number of categories (ordinate) and error rate (abscissa). Along each curve (from left to right), each successive point corresponds to training on a larger training set. So, each curve shows error rate and number of categories resulting from training on 10 to the power 2, 3, 4, and 5 training samples.

FA. GA was evaluated with γ values that yielded similar numbers of categories as FA, $\gamma = 2.0$, and $\gamma = 4.0$. The input data were renormalized to have unit variance in each dimension for GA. Also, for

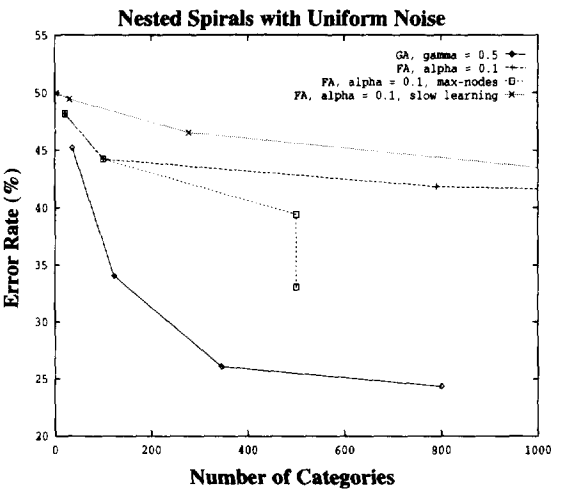

FIGURE 11. Nested spirals with uniformly distributed noise: test results.

TABLE 1

Table Shows Number of Categories and Final Classification Results, for Frey–Slate Letter Image Recognition Problem (with Full Set of 16 Features), of Nearest-neighbor (NN \k), Fuzzy ARTMAP (FA), and Gaussian ARTMAP (GA). FA was Run with $\alpha = 1.0$ and $\alpha = 0.1$, for which it Equilibrated after 7 and 13 Epochs, Respectively. GA was Run with $\gamma = 2.0$ and $\gamma = 4.0$ for 20 Epochs. For each Parameter Setting, FA and GA were Trained Independently five different times. The five results were Averaged to Produce 1-Voter Results, and were Combined using FA's and GA's Respective Voting Methods to Produce the Five-Voter Results

Classifier	Parameter	Number of Voters	Number of Categories	Percent Correct
NN	—	—	16,000	95.80
FA	$\gamma = 1.0$	1	1035	91.90
FA	$\alpha = 0.1$	1	807	86.53
GA	$\gamma = 2.0$	1	1044	93.97
GA	$\gamma = 4.0$	1	975	93.71
FA	$\alpha = 1.0$	5	5175	94.85
FA	$\alpha = 0.1$	5	4035	91.73
GA	$\gamma = 2.0$	5	5218	95.95
GA	$\gamma = 4.0$	5	4876	95.30

comparison, the nearest-neighbor (NN) classifier was evaluated.

Table 1 summarizes the final results using all 16 features. FA with $\alpha = 0.1$ equilibrated after seven epochs, and with $\alpha = 1.0$ after 13 epochs. Without voting, GA's classification rates with either value of γ are about 2% better than FA's best rate, with $\alpha = 1.0$. With voting, FA gains ground on GA, so that GA's rate with $\gamma = 2.0$ is about 1% better than FA's best rate, and GA's rate with $\gamma = 4.0$ is about 0.5% better. The only result superior to the nearest-neighbor (NN) rate of 95.80% is the GA result (with $\gamma = 2.0$ and voting) of 95.95%.

Table 2 summarizes the final results with the reduced set of 11 features. With the reduced feature set, none of the systems equilibrated before 20 epochs. Note that FA's classification rate on the reduced feature set is higher than on the full set, while its number of categories remains roughly constant. GA's classification rate, on the other hand, only improves slightly, but the number of categories it uses decreases substantially. Thus, additional features with low discriminative value had a different effect on the two systems: they caused FA's classification rate to decrease while its number of categories remained stable, and caused GA's classification rate to remain stable while its number of categories increased.

With the reduced feature set, GA achieves a higher classification rate without voting than FA, while using fewer categories. With voting, GA (with $\gamma = 2.0$) and FA (with $\alpha = 1.0$) both achieve a rate

TABLE 2

Table Shows Number of Categories and Final Classification Results, for Frey–Slate Letter Image Recognition Problem (with Reduced Set of 11 Features), of Nearest Neighbor (NN), Fuzzy ARTMAP (FA), and Gaussian ARTMAP (GA). FA and GA were Trained for 20 Epochs without Equilibrating

Classifier	Parameter	Number of Voters	Number of Categories	Percent Correct
NN	—	—	16,000	96.55
FA	$\alpha = 1.0$	1	1062	94.03
FA	$\alpha = 0.1$	1	800	89.54
GA	$\gamma = 2.0$	1	844	94.55
GA	$\gamma = 4.0$	1	768	94.10
FA	$\alpha = 1.0$	5	5312	95.82
FA	$\alpha = 0.1$	5	4001	93.16
GA	$\gamma = 2.0$	5	4208	95.98
GA	$\gamma = 4.0$	5	3838	95.55

of nearly 96% correct, however GA uses fewer categories. The best result is obtained by NN, with 96.55% correct.

Figure 12 plots the test error rates and number of categories produced by GA and FA, across all 20 training epochs, for the full feature set. The abscissa ranges from 700 to 1050 categories, and the ordinate ranges from 0% to 17% error. Note that GA and FA have similar error rates following one training epoch, but with subsequent epochs GA's error rate decreases more than FA's, so that GA ends up with a lower error rate.

The performance when tested on the training data

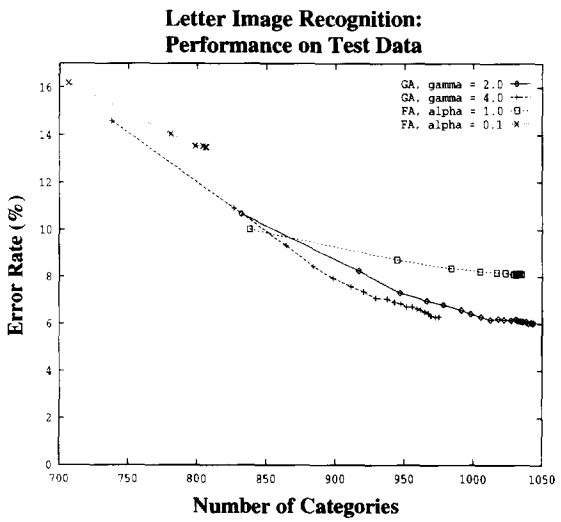

FIGURE 12. GA and FA number of categories (ordinate) and error rates (abscissa) on test data of letter image recognition task, with full set of 16 features. Each successive point along a curve (from left to right) corresponds to a training epoch. Results are averaged across five independently trained systems. The same train/test partition of 16,000 train and 4000 test samples is used on all five runs.

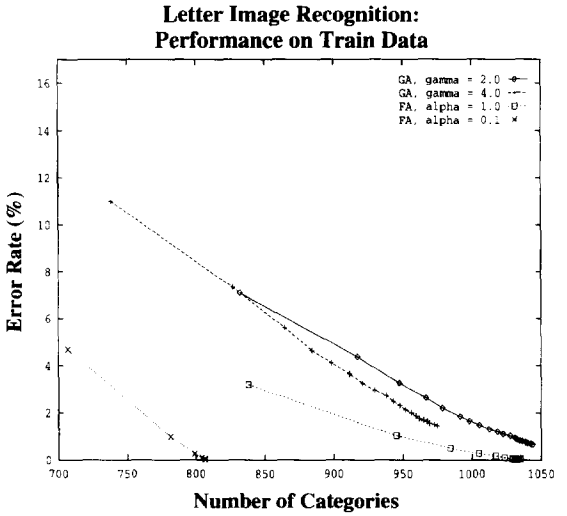

FIGURE 13. GA and FA error rates on train data of letter image recognition task, with full set of 16 features.

is quite different. Figure 13 shows that FA has a much lower error rate on the training data than GA. The difference between the error rate on training data and testing data is particularly striking for FA with $\alpha = 0.1$. FA has a tendency, especially when α is small, to overlearn its training data and generalize poorly to its testing data.

Figure 14 plots the test error rates with voting. Note that although both systems benefit from voting, FA benefits more than GA, achieving error rates nearly as low as those for GA. In fact, for the first couple epochs, FA (with $\alpha = 1.0$) achieves a lower error rate than GA, but GA, for both settings of γ, overtakes FA's best results with sufficient training.

With the reduced feature set, the shape of the error rate curves are quite similar to those for the full

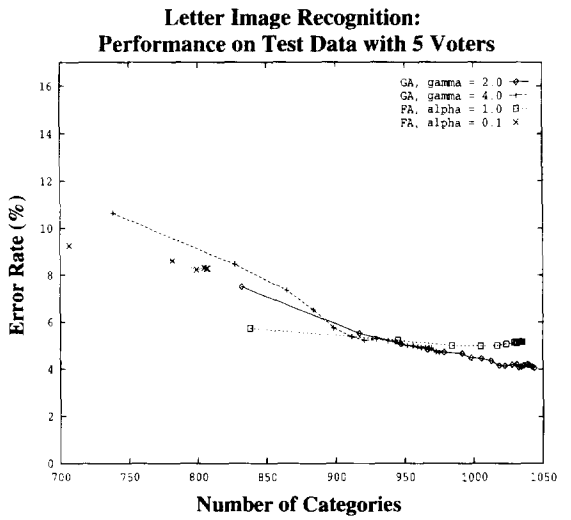

FIGURE 14. GA and FA error rates, with five voters, on test data of letter image recognition task with full set of 16 features. Ordinate represents average number of categories used by the five voting systems.

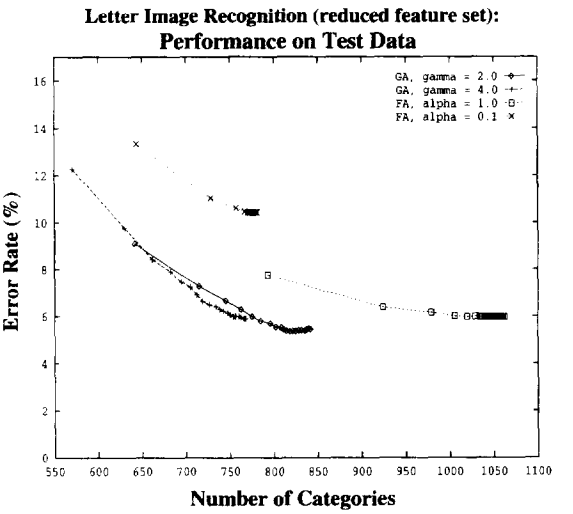

FIGURE 15. GA and FA number of categories (ordinate) and error rates (abscissa) on test data of letter image recognition task, with reduced set of 11 features.

feature set, although the curves are shifted somewhat. Figure 15 plots the test error rates and number of categories produced, in which the abscissa ranges from 550 to 1100 categories, and the ordinate ranges from 0% to 17% error. With respect to the full feature set results plotted in Figure 12, GA's curves are primarily shifted to the left, and FA's curves primarily shifted downward. The training curves and voting curves in Figures 16 and 17 are similarly shifted. In Figure 17, note that FA, with voting, achieves its highest classification rate after a couple of epochs, while GA requires several epochs to achieve a similarly high rate.

5.4. Speaker Independent Vowel Recognition

GA and FA were also evaluated on a speaker

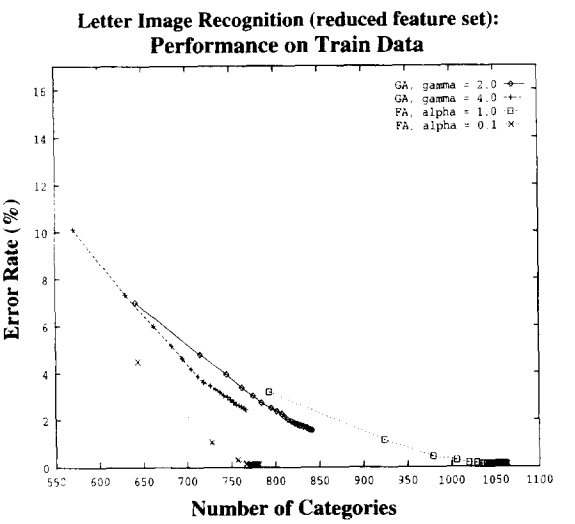

FIGURE 16. GA and FA error rates on train data of letter image recognitiont task, with reduced set of 11 features.

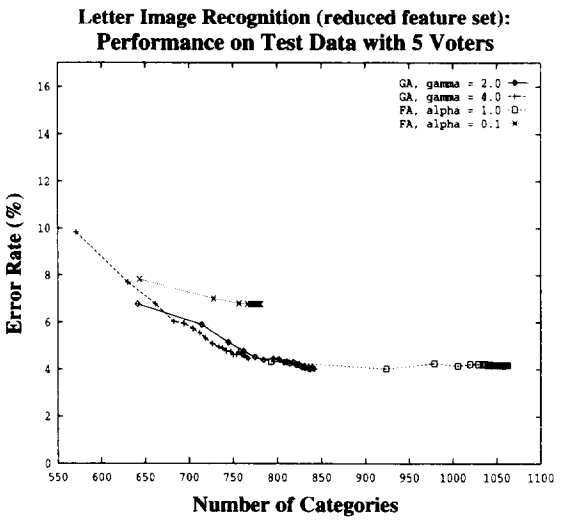

FIGURE 17. GA and FA error rates, with five voters, on test data of letter image recognition task with reduced set of 11 features. Ordinate represents average number of categories used by the five voting systems.

independent vowel recognition database, for which results of many other neural classifiers are available for comparison. The data were collected by Deterding (1989), who recorded examples of the 11 steady-state vowels of English spoken by 15 speakers. The vowel data are electronically available from the Carnegie-Mellon University connectionist benchmark collection (see Fahlman, 1993).

Table 3 shows the ASCII approximation to the International Phonetic Association (IPA) symbol for the 11 vowel sounds, and the word in which each was recorded. Each word was spoken once by each of the 15 speakers, seven of whom were female and eight male. The speech signals were low pass filtered at 4.7 kHz and then digitized to 12 bits with a 10-kHz sampling rate. Twelth-order linear predictive analysis was carried out on six 512 sample Hamming windowed segments from the steady part of the vowel. The reflection coefficients were used to calculate 10 log area parameters, giving a 10-dimensional input space. Each speaker thus yielded six samples of speech from the 11 vowels, resulting in 990 samples from the 15 speakers.

Robinson (1989) used this data to investigate several types of neural network algorithms. He partitioned the data into 528 samples for training, from four male and four female speakers, and 462 samples for testing, from the remaining four male and three female speakers. He examined several classifiers: a single-layer perceptron, multilayer networks with sigmoidal, Gaussian, and quadratic activation functions, a modified Kanerva model, radial basis networks, and nearest neighbor. Robinson's results are shown in the first part of Table 4. Each result is based on a single run using random starting weights.

TABLE 3
Words Used for Recording Vowels to Create Vowel Recognition Database (Adapted from Robinson, 1989)

Vowel	Word	Vowel	Word
i	Heed	O	Hod
I	Hid	C:	Hoard
E	Head	U	Hood
A	Had	u:	Who'd
a:	Hard	3:	Heard
Y	Hud		

All results are reported following about 3000 epochs, although peak performance may have been obtained with fewer. In most cases, performance peaked at around 250 (54%) correct, after which it degraded by different amounts. See Robinson (1989) for more details.

Much better results on this database were obtained with a classifier called growing cell structures (GCS), which is similar to a radial basis network (Fritzke, 1994). With GCS, each radial basis function corresponds to a cell in a self-organizing feature map (Malsburg, 1973; Grossberg, 1976, 1978; Kohonen, 1984), a topological graph of a preconfigured dimensionality, in which learning of input patterns is shared between a cell and its nearest neighbors in the graph. Each cell's basis function (e.g., a Gaussian distribution) has a standard deviation determined by the mean length of graph edges connected to that cell. Self-organization of the RBF layer and the supervised adaptation of output weights are done in parallel during training, and training error is used to determine where to insert new cells.

Like GA, therefore, GCS incrementally creates the appropriate number of hidden units to adequately perform the input/output mapping, using smooth basis functions. Unlike GA, however, GCS does not use simple local update rules which could be readily implemented in parallel hardware. Insertion of new cells into the graph is a particularly complex nonlocal operation (see Fritzke, 1994, p. 1447). The GCS results of Fritzke (1994) are summarized in the second section of Table 4, in which results from three runs of a three-dimensional GCS network, and two runs of a five-dimensional GCS network, are averaged.

The results of FA and GA are shown in the third and fourth sections of Table 4. Presumably because the database is so small, FA and GA produced results with rather high variability on independent runs, due to different orderings of the training data. Therefore, the results of five different voting runs were averaged. Each voting run in turn consisted of five runs for each of the voters. Unlike the other neural network systems, FA and GA both trained very rapidly. FA achieved its maximum performance after just two

TABLE 4

Table Shows (Rounded to Nearest Integer) Number of Training Epochs, Number of Categories, Number of Correctly Classified Samples, and Percentage Correct Classification of Speaker Independent Vowel Recognition Task for Several Classification Methods from three Different Studies. The First Section Shows the Results Obtained by Robinson on many Neural Network Classifiers, as well as the Nearest Neighbor Classifier. The Ground Section Shows the Average Results Obtained by Fritzke with a Three-dimensional and Five-dimensional GCS Network. The Third Section Shows the Results Obtained, with and without Voting, by FA with $\alpha = 1.0$, and $\alpha = 0.1$. The Fourth Section Shows Results Obtained, with and without Voting, by GA with $\gamma = 2.0$, and $\gamma = 4.0$

Classifier	Number of Epochs	Number of Categories (Hidden Units)	Correctly Classified	Percent Correct
Single-layer perceptron	3000	—	154	33
Multilayer perceptron	3000	88	234	51
Multilayer perceptron	3000	22	206	45
Multilayer perceptron	3000	11	203	44
Modified Kanerva Model	3000	528	231	50
Modified Kanerva Model	3000	88	197	43
Radial basis function	3000	528	247	53
Radial basis function	3000	88	220	48
Gaussian node network	3000	528	252	55
Gaussian node network	3000	88	247	53
Gaussian node network	3000	22	250	54
Gaussian node network	3000	11	211	47
Square node network	3000	88	253	55
Square node network	3000	22	236	51
Square node network	3000	11	217	50
Nearest Neighbor	1	528	260	56
Three-dimensional GCS	80	159	292	63
Five-dimensional GCS	80	166	307	66
FA, $\alpha = 1.0$	2	66	236	51
FA, $\alpha = 0.1$	2	56	229	49
FA, $\alpha = 1.0$, 5 voters	10	329	246	53
FA, $\alpha = 0.1$, 5 voters	10	279	244	53
GA, $\gamma = 2.0$	4	56	262	57
GA, $\gamma = 4.0$	20	55	269	59
GA, $\gamma = 2.0$, 5 voters	20	279	287	62
GA, $\gamma = 4.0$, 5 voters	100	273	292	63

epochs for both settings of α. GA achieved its maximum performance after just four epochs with $\gamma = 2.0$, and after 20 epochs with $\gamma = 4.0$. Without voting, GA achieved better results (57% and 59% correct for $\gamma = 2.0$ and $\gamma = 4.0$ respectively) than FA with or without voting, better than all neural networks tested by Robinson, and better than the nearest-neighbor classifier. With $\gamma = 2.0$, this result was obtained after only four training epochs, using 56 categories. With voting, GA achieved classification results comparable to those of the three-dimensional GCS network. With $\gamma = 2.0$, GA achieved 62% correct using 20 net training epochs (four epochs and five voters = 20 epochs) and 279 categories among the five trained systems. By setting γ higher ($\gamma = 4.0$), GA relaxed more slowly and achieved better final classification, with 63% correct using 100 net training epochs and 273 categories.

6. CONCLUSION

A new neural network architecture called Gaussian ARTMAP has been introduced, which is based on the synthesis of a Gaussian classifier and an ARTMAP neural network. In comparison to another ARTMAP architecture called fuzzy ARTMAP, Gaussian ARTMAP has more complex learning rules and choice and match functions, yet retains fuzzy ARTMAP's attractive fast learning and parallel computing properties. Gaussian ARTMAP has been shown to generalize better to test data and to be more resistant to noisy training data than fuzzy ARTMAP. Gaussian ARTMAP has also achieved results on a vowel recognition database which are better than those of many standard neural network classifiers, and nearly as good as the best previously published results, which were achieved with the growing cell structures network (Robinson, 1989; Fritzke, 1994).

Resistance to noise is important for incremental learning systems, such as fuzzy ARTMAP, Gaussian ARTMAP, and growing cell structures, which create a sufficient number of categories, or hidden units, to perform a multidimensional mapping of their training data. In a real-world task, such as incremental learning by a mobile robot, a network

would receive a virtually infinite number of training samples, which may be very noisy, from sensor inputs. For this reason, it is not sufficient to demonstrate that a network avoids category proliferation when trained for several epochs on the same small data set. In this paper, Gaussian ARTMAP was trained on large, noisy data sets and achieved very good classification with only a moderate proliferation of categories.

An area for future investigation is how best to initialize the variance of Gaussian ARTMAP categories. Currently, they are initialized with a constant standard deviation of γ in (12). Using a large γ results in slower training with fewer categories, while using a small γ results in faster training with more categories. In terms of classification rate, an optimal γ exists for each data set, but this value varies between data sets. It might be useful to vary γ over time, perhaps to start training with a large γ, and decrease γ as training progresses.

Another area for investigation is to further restrict the proliferation of categories. One way to do this is to prune categories which are chosen infrequently and/or have low predictive value, as was done by Carpenter and Tan (1995) using fuzzy ARTMAP for the learning of IF-THEN rules in medical database classification. Similar extraction of IF-THEN rules based on GA's category means, variances, and a priori probabilities is also an area for investigation.

REFERENCES

Broomhead, D. S., & Lowe, D. (1988). Multivariable functional interpolation and adaptive networks. *Complex Systems*, **2**, 321–355.

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

Carpenter, G. A., & Tan, A. H (1995). Rule extraction, From neural architecture to symbolic representation. *Connection Science*, **7**, 3–27.

Carpenter, G. A., Grossberg, S., & Reynolds, J. (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., Grossberg, S., & Iizuka, K. (1992a). Comparative performance measures of fuzzy ARTMAP, learned vector quantization, and back propagation for handwritten character recognition. *Proceedings IJCNN-92*, **I**, 794–799.

Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J., & Rosen, D. B. (1992b). 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. (1995). A Fuzzy ARTMAP nonparametric probability estimator for nonstationary pattern recognition problems. *IEEE Transactions on Neural Networks*, **6**, 1330–1336.

Courant, R., & Hilbert, D. (1962). *Methods of mathematical physics*. London: Interscience.

Deterding, D. H. (1989). *Speaker normalisation for automatic speech recognition*. PhD thesis, University of Cambridge.

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

Fahlman, S. E. (1993). *CMU benchmark collection for neural net learning algorithms*. Carnegie Mellon University, School of Computer Science [machine-readable data repository], Pittsburgh.

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

Fritzke, B. (1994). Growing cell structures—A self-organizing network for unsupervised and supervised learning. *Neural Networks*, **7**, 1441–1460.

Grossberg, S. (1976). Adaptive pattern classification and universal recoding, I: Parallel development and coding of neural feature detectors. *Biological Cybernetics*, **23**, 121–134.

Grossberg, S. (1978). A theory of human memory: Self-organization and performance of sensory-motor codes, maps, and plans. In R. Rosen & F. Snell (Eds.), *Progress in theoretical biology* (Vol. 5, pp. 233–374). New York: Academic Press.

Kohonen, T. (1984). *Self-organization and associative memory*. New York: Springer-Verlag.

Kruschke, J. K. (1992). ALCOVE: An exemplar-based connectionist model of category learning. *Psychological Review*, **99**, 22–44.

Malsburg, C. von der (1973). Self-organization of orientation sensitive cells in the striate cortex. *Kybernetik*, **14**, 85–100.

Poggio, T., & Girosi, F. (1989). A theory of networks for approximation and learning. *A.I. Memo No. 1140*.

Popat, K., & Picard, R. W. (1993). *Novel cluster-based probability model for texture synthesis, classification, and compression*. MIT Media Laboratory Perceptual Computing Group Technical Report 234.

Powell, M. J. D. (1987). Radial basis functions for multivariable interpolation: a review. In J. C. Mason & M. G. Cox (Eds.) *Algorithms for approximation*. Oxford: Clarendon Press.

Robinson, A. J. (1989). *Dynamic error propagation networks*. PhD thesis, Cambridge University.

Stone, C. J. (1982). Optimal global rates of convergence for nonparametric regression. *Ann. Stat.*, **10**, 1040–1053.

Tikhonov, A. N., & Arsenin, V. Y. (1977). *Solutions of ill-posed problems*. Washington, D.C.: W. H. Winston.

NOMENCLATURE

I	input vector for a training or testing sample
M	number of features per sample
j	index of ART category
J	index of chosen ART category
$T_j(I)$	fuzzy ART choice function
$\arg\max_j(T_j(I))$	picks index j for which $T_j(I)$ is maximum
$w_J := I \wedge w_J$	same as: $w_J^{(\text{new})} = I \wedge w_J^{(\text{old})}$
$\lvert I \rvert$	length of I: $\sum_{i=1}^{M} I_i$

w_j	weight vector for fuzzy ART category j
α	fuzzy ART choice parameter
$\rho, \bar{\rho}$	ART vigilance parameter, and baseline vigilance parameter
μ_j	mean vector for Gaussian ART category j
σ_j	standard deviation vector for Gaussian ART category j
n_j	count scalar for Gaussian ART category j
$P(j\|I)$	a posteriori probability of category j given input I
$p(I\|j)$	conditional density of input I given category j
$P(j)$	a priori probability of category j
N	number of ART categories
$g_j(I)$	Gaussian ART choice function
$g'_j(I)$	Gaussian ART match function
γ	Gaussian ART standard deviation initialization parameter
K'	ARTMAP class prediction of chosen category J
$\Omega(\,)$	ARTMAP function which maps categories to a class prediction
V	number of ART systems which are combined via voting