Available online at www.sciencedirect.com
SCIENCE DIRECT
Neural Networks 19 (2006) 446–468
Neural Networks
www.elsevier.com/locate/neunet

Boosted ARTMAP: Modifications to fuzzy ARTMAP motivated by boosting theory

Stephen J. Verzi [a], Gregory L. Heileman [b,*], Michael Georgiopoulos [c]

[a] Computer Science Department University of New Mexico, Albuquerque, NM 87131, USA
[b] Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM 87131, USA
[c] Department of Electrical and Computer Engineering, University of Central Florida, Orlando, FL 32816, USA

Received 28 June 2004; accepted 10 August 2005

Abstract

In this paper, several modifications to the Fuzzy ARTMAP neural network architecture are proposed for conducting classification in complex, possibly noisy, environments. The goal of these modifications is to improve upon the generalization performance of Fuzzy ART-based neural networks, such as Fuzzy ARTMAP, in these situations. One of the major difficulties of employing Fuzzy ARTMAP on such learning problems involves over-fitting of the training data. Structural risk minimization is a machine-learning framework that addresses the issue of over-fitting by providing a backbone for analysis as well as an impetus for the design of better learning algorithms. The theory of structural risk minimization reveals a trade-off between training error and classifier complexity in reducing generalization error, which will be exploited in the learning algorithms proposed in this paper. Boosted ART extends Fuzzy ART by allowing the spatial extent of each cluster formed to be adjusted independently. Boosted ARTMAP generalizes upon Fuzzy ARTMAP by allowing non-zero training error in an effort to reduce the hypothesis complexity and hence improve overall generalization performance. Although Boosted ARTMAP is strictly speaking not a boosting algorithm, the changes it encompasses were motivated by the goals that one strives to achieve when employing boosting. Boosted ARTMAP is an on-line learner, it does not require excessive parameter tuning to operate, and it reduces precisely to Fuzzy ARTMAP for particular parameter values. Another architecture described in this paper is Structural Boosted ARTMAP, which uses both Boosted ART and Boosted ARTMAP to perform structural risk minimization learning. Structural Boosted ARTMAP will allow comparison of the capabilities of off-line versus on-line learning as well as empirical risk minimization versus structural risk minimization using Fuzzy ARTMAP-based neural network architectures. Both empirical and theoretical results are presented to enhance the understanding of these architectures.

1. Introduction

An important performance measure of a machine-learning algorithm is its generalization capability; that is, for a hypothesis output by the learning algorithm, how well does it predict randomly chosen examples? The standard problem is as follows: A learning algorithm is supplied with a pre-chosen set of labeled training examples, $S = \{(\mathbf{x}_1, \mathbf{y}_1),\ldots,(\mathbf{x}_N, \mathbf{y}_N)\}$, that it will use in order to output a hypothesis. The chosen hypothesis should perform well on the training data or there would be no reason for employing it. However, how well it can predict previously unseen examples constitutes a measure of how well the learning algorithm can generalize, from the training set, to the underlying distribution of example data. The focus of learning in this paper is on a particularly difficult situation in which the training data contains seemingly conflicting information. The conflicting information may be due to naturally overlapping pattern class distributions or to noise injected during data acquisition. In these types of learning problems, a learning algorithm must be flexible enough to deal with the conflicting information when producing a hypothesis, so that it can predict unseen examples with a high degree of accuracy. These types of learning problems are characteristic of many real-world learning situations.

The research in this paper will focus on adaptive resonance theory (ART) neural networks, specifically, Fuzzy ARTMAP (Carpenter et al., 1992). Fuzzy ARTMAP is an example of a constructive neural network model in that it allows nodes to be added as necessary during training. The growth potential of Fuzzy ARTMAP is similar to that of other constructive learning algorithms such as decision tree learners; they are all allowed to grow as necessary to suite a particular set of training data (Kearns and Mansour, 1995). The fact that Fuzzy

* Corresponding author.
E-mail address: heileman@ece.unm.edu (G.L. Heileman).

0893-6080/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2005.08.013

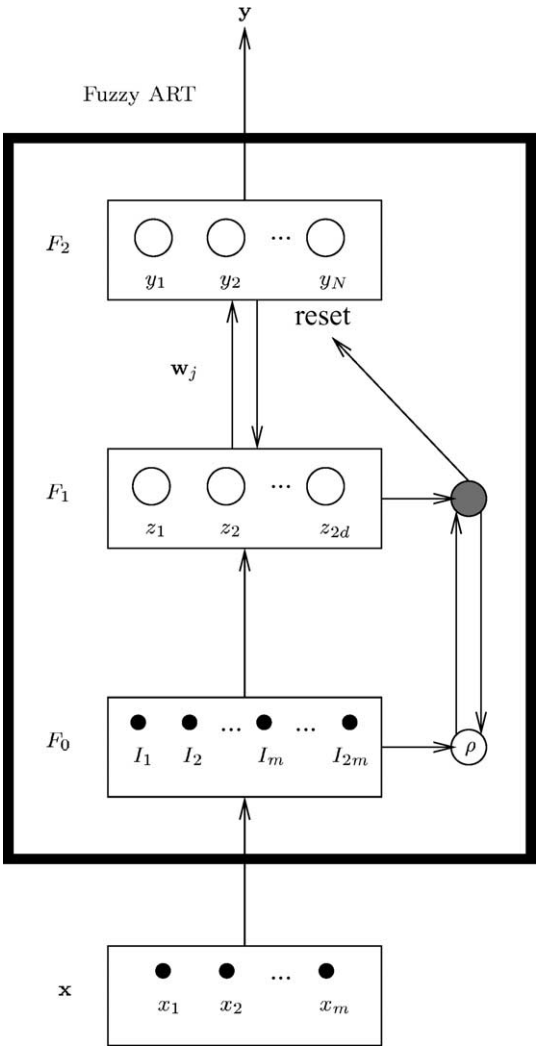

Fig. 1. The Fuzzy ART architecture.

ARTMAP is a constructive algorithm is certainly an advantage, but in situations where there is noise or inherent overlap in the training data it can produce networks that are larger than necessary, resulting in a loss of generalization performance. In these situations Fuzzy ARTMAP can over-fit the training data.

Two distinct sources inside Fuzzy ARTMAP that can contribute to over-fitting will be discussed. In the Fuzzy ART network of Fig. 1, a single adjustable parameter, called the vigilance parameter ρ, is used to control the spatial extent of all clusters simultaneously. In Fig. 2(a), the regions of attraction (or spatial extents) of two Fuzzy ART clusters in $\Re^2$ at a vigilance value of 0.0 are shown. Each F_2 node of a Fuzzy ART neural network maps to a roughly hyper-rectangular region in $\Re^m$. Thus, in Fig. 2(a), the white dashed lines indicate the hyperbox region of each F_2 node. These hyperboxes correspond to training data points that have actually been 'seen' by the learner. In Fuzzy ART with complement coding, training data points will always lie inside of these hyperboxes. In Fig. 2(a), the points inside the unit square have been colored (either black or gray) according to their membership in a cluster or by their closest association to that particular cluster. These colored regions correspond to prediction labels that would be output from a trained Fuzzy ARTMAP neural network, one color for each committed F_2 node representing its own cluster. In Fuzzy ART, the uncommitted node, $\mathbf{y}_N$ in Fig. 1, will attract all points that are too far away from any of the committed clusters. The uncommitted region is shown as white in Fig. 2(a). The actual region covered by a specific F_2 node depends directly upon the current vigilance value. At a higher vigilance value of 0.7, the spatial extents of the two clusters change, rather drastically, as shown in Fig. 2(b). Note that the uncommitted cluster node now covers more of the space in Fig. 2(b). An important fact concerning Fuzzy ART and its single vigilance parameter is that the two white dashed hyperbox regions shown in Fig. 2(a) and (b) cannot be precisely represented with only two F_2 nodes at a single vigilance value.

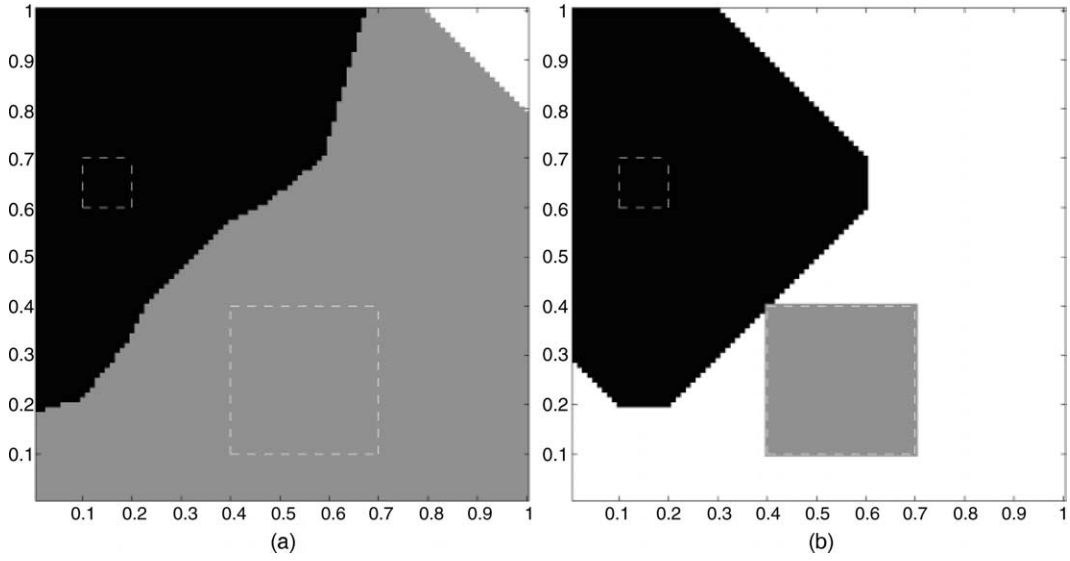

Fig. 2. Regions of attraction for two Fuzzy ART cluster templates at (a) $\rho = 0.0$, and (b) $\rho = 0.7$.

During learning in Fuzzy ARTMAP, the vigilance value of the *A*-side Fuzzy ART module is allowed to change as necessary to fit a particular set of training data, and it is this feature that allows the match tracking, shown in Fig. 3, to work correctly. This feature can also make it difficult to analyze Fuzzy ARTMAP performance since the sizes of the clusters will shrink and/or grow depending upon the current value of the vigilance. The performance of Structural Boosted ARTMAP, presented later in this paper, allows for much more structured analysis. In fact, this particular neural network is a proof of concept for the universal function approximation capabilities of Fuzzy ART-based neural networks (Verzi, 2003; Verzi et al., 2003)

The MAP field of Fuzzy ARTMAP, shown in Fig. 3, and how it is used to connect the two Fuzzy ART sub-modules can also contribute to over-fitting in training. The Fuzzy ARTMAP MAP field has a user-supplied input parameter, ρ^{AB}, which might be used for controlling the crispness of association between the *A*-side and *B*-side Fuzzy ART sub-modules, except that the weighted links, $\mathbf{w}^{AB}$, are always either 0 or 1. Crispness, here, refers to the degree of fuzzy association between *A*-side and *B*-side F_2 nodes. The MAP field weights of Fuzzy ARTMAP are themselves always either one or zero which sets up a totally crisp association between the Fuzzy ART sub-modules. With such crisp weights, ρ^{AB} (MAP field vigilance) cannot be easily used to control the association crispness. The MAP field vigilance parameter takes on values between zero and one, and so it can only be used to allow or disallow associations that are themselves either zero or one. Clearly the Fuzzy ARTMAP MAP field does not represent a fuzzy or a probabilistic relationship between the data and associated labels. There are existing modifications of Fuzzy ARTMAP in the literature, including Gaussian ARTMAP (Williamson, 1996), ART-EMAP (Carpenter and Ross, 1995), PROBART (Marriott and Harrison, 1995) and Micro ARTMAP (Gómez-Sánchez et al., 2002), that attempt to address this issue by a using probabilistic approach, and these will be discussed in more detail later in this paper.

Due to the lack of flexibility of the single Fuzzy ART vigilance parameter and the crispness of the Fuzzy ARTMAP MAP field, more clusters than necessary are constructed during learning in some situations. When too many F_2 nodes are generated during learning, it can result in a solution that is less general, causing a higher generalization error. The number of F_2 nodes used during training is a good measure of the complexity of the Fuzzy ARTMAP neural network. During learning in Fuzzy ARTMAP, it is appropriate to apply Occam's razor wherein 'all things being equal, a solution with less resources (complexity) is preferred over a more complex one (Blumer et al., 1987)'.

This paper focuses on addressing these two weaknesses of Fuzzy ARTMAP by proposing extensions to the Fuzzy ART and Fuzzy ARTMAP neural network architectures that are motivated by boosting theory. Iterative boosting can be described in general by two processes. First, control generalization error during each iteration so that more than

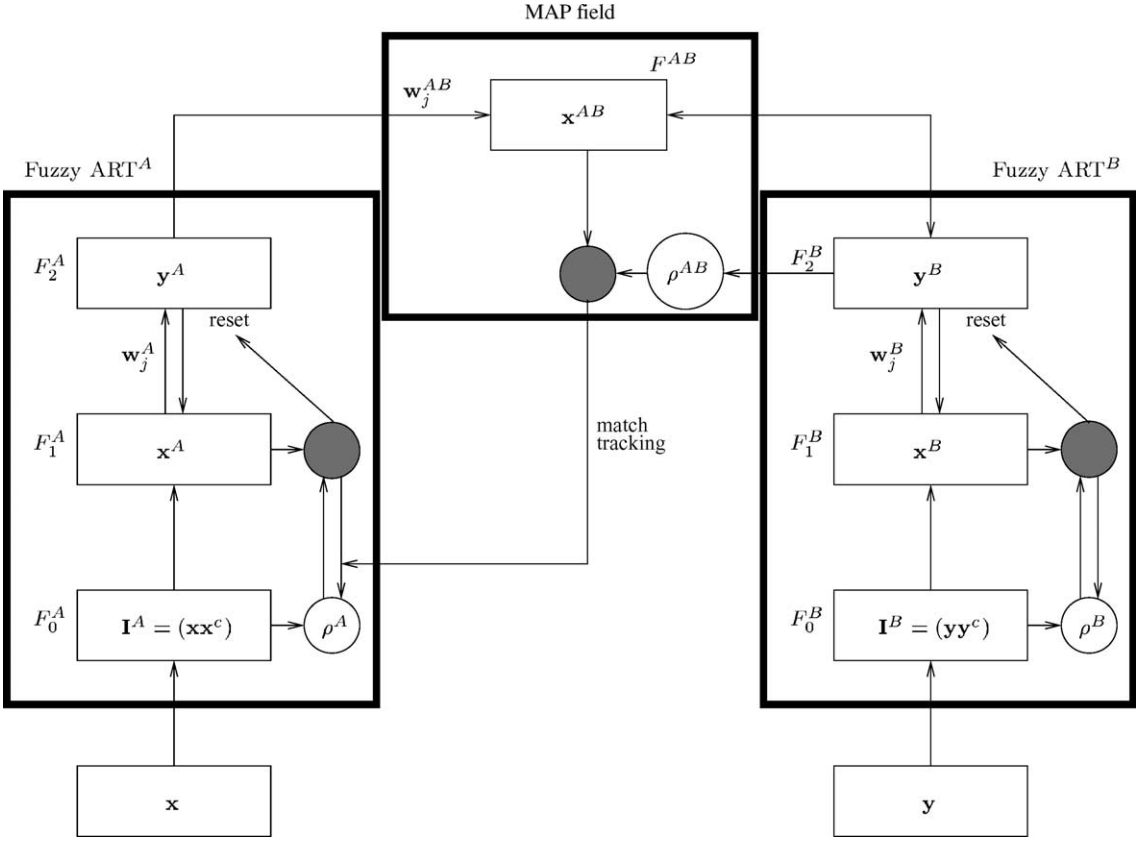

Fig. 3. The Fuzzy ARTMAP architecture.

half of the training samples are correctly classified. Second, focus the current iteration of learning by using training examples which are difficult to classify in previous iterations. In Fuzzy ARTMAP operation, there is no notion of an 'iteration' due to its use of on-line learning. However, Fuzzy ARTMAP does focus in on hard to classify training examples, but it does not attempt to control generalization error in these situations. Instead Fuzzy ARTMAP attempts to directly eliminate generalization error by dealing with difficult training data as soon as it is encountered during training, which is appropriate for on-line learning.

The first modification proposed in this paper involves a generalization of Fuzzy ART into Boosted ART where each F_2 node (representing a data cluster) has its own vigilance parameter for separately controlling its spatial extent. As in the Fuzzy ART architecture shown in Fig. 1, the Boosted ART architecture maintains an extra F_2 node as an uncommitted node. In Boosted ART, this node maintains the value of the baseline vigilance, ρ. Newly constructed or committed cluster nodes in the Boosted ART neural network receive the base-line vigilance value. During Boosted ART learning, the vigilance values of committed F_2 nodes are allowed to change as necessary. In Boosted ART, if each of the F_2 node vigilance values are set and maintained at the same value as the base-line vigilance at all times, then Boosted ART reduces precisely to Fuzzy ART.

The addition of separate vigilance parameters for each F_2 node in Boosted ART generalizes Fuzzy ART by allowing increased control over the spatial extents of each cluster formed during learning. The Boosted ART F_2 node is more flexible than the Fuzzy ART F_2 node. In fact, the two hyperboxes from Fig. 2 can now be precisely represented by two F_2 nodes in Boosted ART with separate vigilance values, as can be seen in Fig. 4(a). In general, Boosted ART can require fewer F_2 nodes to represent the same measurable space as Fuzzy ART.

The second modification proposed in this paper involves a generalization of the Fuzzy ARTMAP and PROBART MAP fields into the Boosted ARTMAP MAP field. The Fuzzy ARTMAP MAP field is modified to create the Boosted ARTMAP MAP field, similar to PROBART (Marriott and Harrison, 1995). In PROBART, each weighted link between A-side and B-side F_2 nodes is allowed to increase from zero to some value K, where K is the number of associations created during learning. In the Boosted ARTMAP MAP field this accumulated association information is used to directly estimate the amount of error allowed in the associated links of the MAP field. Given the law of large numbers, the estimate of the error in association links becomes more accurate as more training data is seen (Devroye et al., 1996; Vidyasagar, 1997). The use of the Boosted ARTMAP MAP field is important because it extends the usefulness of Fuzzy ARTMAP to cases where it previously performed poorly.

Boosted ARTMAP adds an additional input parameter for the desired error tolerance allowed in the MAP field. This parameter was designed to be used in place of Fuzzy ARTMAP's MAP field vigilance parameter. Boosted ARTMAP can be made to function precisely as Fuzzy ARTMAP by setting the MAP field error tolerance to 0.0, and it can be made to operate precisely as PROBART, with maximum frequency prediction, by setting the MAP field error tolerance to 1.0. It can also be used in situations where Fuzzy ARTMAP would perform poorly, and yet, Boosted ARTMAP will be shown to perform much better in these situations. Determining an appropriate value for the MAP field error tolerance is an open issue, but a simple binary search between the values of 0 and 0.5 works fairly well on all learning problems we have tested. If some information about the amount of overlap or noise is known a priori, then a good error tolerance value can be more easily determined. This issue will be discussed in more detail in the empirical results section below. The Boosted ARTMAP MAP field is used very effectively in learning situations involving conflicting label information, which may be due to data classification overlap or noise. In short, Boosted ARTMAP can perform exactly as Fuzzy ARTMAP

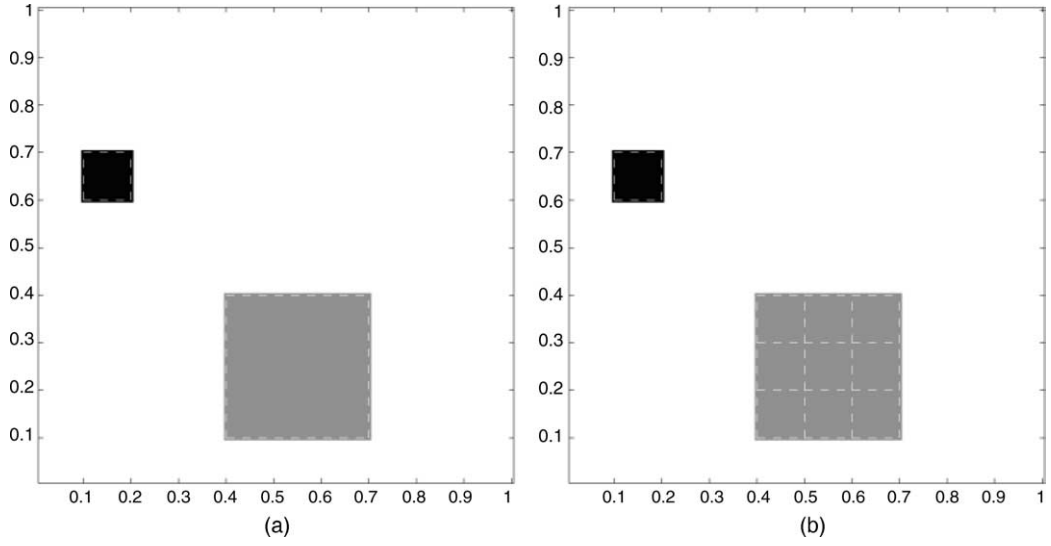

Fig. 4. The spatial extents of (a) two Boosted ART cluster templates at $\rho_1 = 0.9$ and $\rho_2 = 0.7$ and (b) ten Fuzzy ART cluster templates at $\rho = 0.9$.

in situations where Fuzzy ARTMAP does well, but it can outperform Fuzzy ARTMAP in situations involving conflicting training data.

It is important to note that in the strict sense of the definition, Boosted ARTMAP may not be performing boosting. That is, in order to create an on-line learning algorithm that reduces to Fuzzy ARTMAP in the zero noise case, a natural first step is an approximation of what boosting attempts to achieve, leading to Boosted ARTMAP. It is an open question as to whether a learning algorithm that formally achieves boosting in this setting is possible.

Another architecture presented in this paper is Structural Boosted ARTMAP, which makes use of both Boosted ART and the Boosted ARTMAP MAP field. Structural Boosted ARTMAP employs structural risk minimization and off-line learning while demonstrating some of the capabilities of both Boosted ART and Boosted ARTMAP. All of the other Fuzzy ARTMAP-based neural networks mentioned in this paper use on-line learning with empirical risk minimization. The empirical results in Section 6 will show that the use of structural risk minimization by Structural Boosted ARTMAP results in less complex neural network solutions on noisy or overlapping class learning problems. The empirical results will also show that Boosted ARTMAP results in less complex neural network solutions on these same learning problems with appropriate parameter selection without directly using structural risk minimization. Structural Boosted ARTMAP will provide a good comparison case for what is achievable using structural risk minimization and off-line learning in comparison with the Fuzzy ARTMAP-based on-line neural networks.

An overview of this paper is as follows. In Section 2, structural risk minimization will be described as well as a particular type of complexity penalty, called Rademacher penalization which will be used to measure the network complexity of all networks used in the empirical results section. Section 3 contains a brief overview of the Fuzzy ART and Fuzzy ARTMAP neural network architectures. Section 4 contains a complete description of both the Boosted ART and Boosted ARTMAP modified neural network architectures, and Section 5 describes Structural Boosted ARTMAP. Section 6 describes empirical results obtained in comparing the architectures proposed in this paper to Fuzzy ARTMAP and other popular modifications of Fuzzy ARTMAP. Finally, Section 7 provides conclusions and directions for future research.

2. Motivation—a machine learning framework

When dealing with learning problems containing classification overlap or noise, it is important to allow and carefully control the amount of training error present during learning. Vapnik (1995, 1998) developed the structural risk minimization machine learning paradigm to study and bound the performance of existing learning algorithms, as well as to help design better learning algorithms with this sensitivity to training error in mind. Devroye et al. (1996); van der Vaart, and Wellner (1996), and Vidyasagar (1997) and others have continued to advance this learning theory to address areas of interest. Among others, Koltchinskii (2001); Lazano, (2000) have used this machine-learning framework to address neural network learning. In this paper, the structural risk minimization machine learning paradigm will be applied to Fuzzy ART-based and Fuzzy ARTMAP-based neural network learning in order to better understand these models' performance in overlapping and/or noisy environments.

In machine learning, two types of risk minimization techniques have been successfully applied. A learner employing *empirical risk minimization* attempts to minimize training error, even at the cost of hypothesis complexity. Empirical risk minimization is appropriate for well-separated classification tasks. A learner employing *structural risk minimization* attempts to take advantage of the trade-off between representation complexity and approximation error in constructing a hypothesis. Structural risk minimization is more general than empirical risk minimization and appropriate for more complex learning situations, such as those involving class overlap, without significantly degrading performance in easier learning situations.

2.1. Visualizing generalization error

The goal of the work described in this paper is to understand the performance of Fuzzy ARTMAP in an attempt to improve upon its generalization error. A useful way of visualizing generalization error is to note that it comes from two sources, depicted in Fig. 5, representation error and approximation error (Hush, 1997). Representation error, also known as bias, is a measure of the difference between the concept being learned, c in Fig. 5, and the best possible hypothesis that a learner can construct, h^* in Fig. 5. Note that a learner's representation determines the composition of its output hypothesis. For example, a learner whose hypothesis representation is made up of unions of squares will not be able to output a circle, but it could approximate the circle with enough squares of different sizes. It is a known fact that Fuzzy ARTMAP will have non-zero representation error when it is used to learn curved boundaries such as a circle (Carpenter et al., 1992).

Representation error is reduced by using a representation space which more closely matches the underlying structure of the learning problem at hand. Alternately, representation error can be reduced by using a constructive learning algorithm which outputs an aggregate hypothesis composed of very

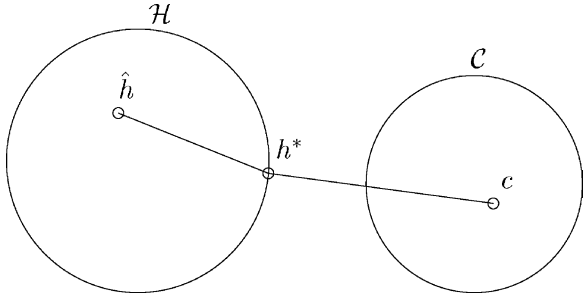

Fig. 5. Types of generalization error in machine learning.

simple components (in terms of their representation) and then allowing the number of components used to increase as necessary. Most Fuzzy ARTMAP-based neural network architectures perform this type of learning. It is also interesting to note that this sort of accumulation of components can be used to achieve boosting (Vapnik, 1998).

Approximation error is a measure of the difference between the actual hypothesis that a learner generates, $\hat{h}$ in Fig. 5, and the best possible hypothesis given its representation, h^* in Fig. 5. Given the training data at hand, it may not be easy for a learner to generate the (theoretical) best possible hypothesis. Note that all Fuzzy ART-based neural network architectures that employ on-line learning will suffer somewhat from approximation error. An off-line learner is not sensitive to the order of presentation of the training data, and thus, this type of learner produces an hypothesis which approximates the entire training set not just a particular permutation of the training data presented in a specific order. Fuzzy ART-based and Fuzzy ARTMAP-based neural networks are limited to approximate the training data in the order that it is presented. One way of detecting this type of error in empirical learning is in the standard deviation across many different presentations of the training data. The empirical results in Section 6 will show that structural risk minimization operated using off-line learning (in Structural Boosted ARTMAP for instance) will not suffer as much from this type of error.

Approximation error is often reduced by limiting the number of hypotheses available to the learning algorithm or by increasing the amount of training data. Note that approximation error is in direct competition with representation error in the case of constructive learning. More hypotheses will give a greater representation capability, but it will be harder in general to find the best of these in terms of approximation error in a particular learning situation (van der Vaart and Wellner, 1996; Vapnik, 1998; Vidyasagar, 1997).

Fig. 6(a) and (b) show situations typical of learning progress where the hypothesis space complexity grows as learning proceeds, representative of Fuzzy ART-based neural networks. In Fig. 6(a) a very desirable progression of learning is shown, where given a training set of size n at a complexity level of N, a hypothesis $\hat{h}_n^N$ is output, and this hypothesis is sufficiently *close* to the target concept c. In Fig. 6(b) the learning has been conducted in the presence of noise or classification overlap. In this case the hypothesis space has become too large, and so the learner has chosen a hypothesis which fits the data according to its algorithm, but which is not *close* to the target concept. Structural risk minimization provides a learning paradigm for addressing the difficulties of constructive learning in overlapping or noisy distributions.

2.2. Structural risk minimization

The goal of *concept learning*, shown in Fig. 6(a), is to find a hypothesis, $\hat{h}^N$, from a class of hypotheses, $\mathcal{H}^N$, with minimal generalization error, i.e.

$$\hat{h}^N = \arg\min_{h\in\mathcal{H}^N} Pr\{h(\mathbf{x}) \neq c(\mathbf{x})\}, \tag{1}$$

where c is the unknown target concept. The well-known estimate of the 'optimal' decision rule is determined by minimizing the empirical risk (Devroye et al., 1996; Vapnik, 1998; van der Vaart and Wellner, 1996; Vidyasagar, 1997). This is called empirical risk minimization learning

$$\hat{h}\hat{N} = \arg\min_{h\in\mathcal{H}^N, N\geq 1} L_n(h) \tag{2}$$

where the measure of empirical risk, $L_n(h)$, is also called training error. Given a particular training set, finding $\hat{h}^N$ precisely may be unfeasible, but it may be possible to find a close approximation to $\hat{h}^N$, called $\hat{h}_n^N$. If $\mathcal{C} \subseteq \mathcal{H}^N$, then the possibility exists for finding the desired concept. In this case, zero training error can be achieved, and employing empirical risk minimization is reasonable assuming efficiency is not compromised. Note that Fuzzy ARTMAP and almost all Fuzzy ARTMAP-based neural networks use empirical risk minimization, and this accounts for their unbounded complexity during training. With structural risk minimization, training error alone is not minimized but rather a combination of training error plus a measure of the learning solution complexity.

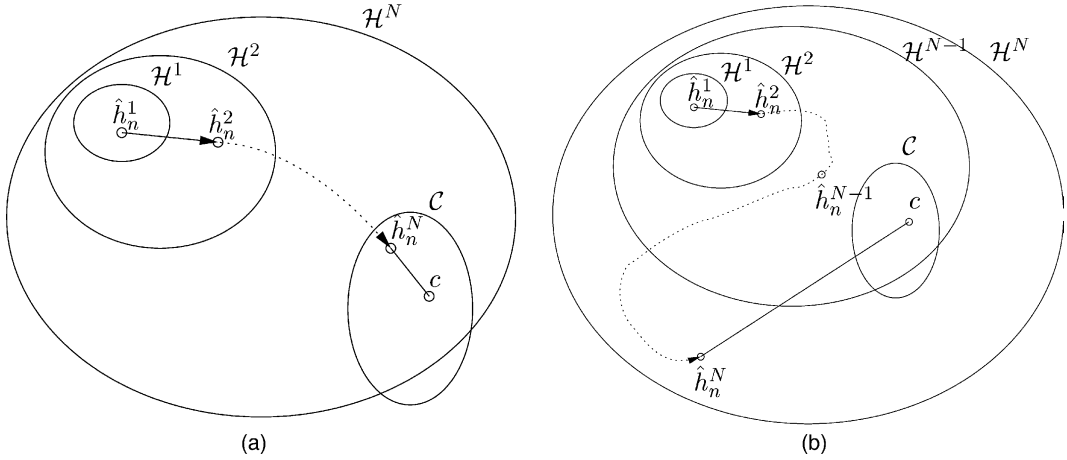

Fig. 6. A learning progression from $\mathcal{H}^1$ to $\mathcal{H}^N$ that is (a) desirable, and (b) overly complex due to some factor such as noise.

2.2.1. Complexity bounding with penalization

The differences between $\mathcal{C}$ and $\mathcal{H}^N$ as well as noise in labeling or overlapping distributions of labels for $\mathcal{C}$ can lead a learner, employing empirical risk minimization, to a sub-optimal solution. Consider that *rote* or *table look-up* learning does not violate empirical risk minimization, but it is not a very general solution. Rote learning is not considered general because there is no mechanism for predicting the classification of new samples, and there is no compaction of previously seen training data. Structural risk minimization was introduced by Vapnik (1995, 1998) and adds a penalty term to the hypothesis risk minimization function of Eq. (2)

$$\hat{h}\hat{N} = \arg\min_{h\in\mathcal{H}^N, N\geq 1}\{L_n(h) + \text{pen}(n; N)\} \tag{3}$$

where pen(n; N) is the penalty computed at complexity N with n training samples. The complexity variable N defines a concentric sieve of levels of complexity including more and more hypotheses as complexity is increased, as is shown in Fig. 6(a) and (b).

It is not hard to see that there is a trade-off between training error and penalization whenever overall generalization error is greater than zero. If $L_n(h)+\text{pen}(n;\ N)>0$, and the training error, $L_n(h)$, is reduced to 0, then there is a non-zero complexity penalty. It is here where empirical risk minimization can perform poorly, and structural risk minimization can be advantageous. The penalty term can be bounded by the Vapnik–Chervonenkis (VC) dimension of the class of learnable hypotheses (Devroye et al., 1996). The VC dimension of a class of hypotheses is one measure of complexity for this set that does not depend upon the underlying distribution of the data (Vapnik and Chervonenkis, 1971).

2.2.2. The Rademacher penalty

The Rademacher penalty was introduced by Koltchinskii as a data-dependent complexity penalty (Koltchinskii, 2001). The Rademacher penalty is computed directly using training data, and thus the inherent distribution of this data is captured as part of the penalization process.

Lazano, (2000) proposes a cleverly simple algorithm for computing the Rademacher penalty for a '0–1'-concept learner. In this method, each training sample ($\mathbf{x}_j$, $\mathbf{y}_j$) is randomly relabeled with probability 0.5. The Rademacher random variables, σ_j, are computed as $\sigma_j=-1$ if $\mathbf{y}_j$ is relabeled, otherwise $\sigma_j=1$ and $\mathbf{y}_j$ is left alone. This new training set is called s_1. A second set of relabeled data is immediately available by reversing all of the labels of s_1, and it is called s_2. Next, the learner is trained using both s_1 and s_2, separately, to produce two hypotheses, h_1 and h_2. The Rademacher penalty is then estimated as

$$\begin{aligned}
\text{pen}(n, h_1) &= \left|\frac{1}{n}\sum_{j=1}^{n}\sigma_j\chi_{\{\mathbf{y}_j\neq h_1(\mathbf{x}_j)\}}(\mathbf{x}_j)\right|, \mathbf{y}_j\in s_1,\\
\text{pen}(n, h_2) &= \left|\frac{1}{n}\sum_{j=1}^{n}\sigma_j\chi_{\{\mathbf{y}_j\neq h_2(\mathbf{x}_j)\}}(\mathbf{x}_j)\right|, \mathbf{y}_j\in s_2,\\
\text{pen}(n, N) &= \max(pen(n, h_1),\ \text{pen}(n, h_2)).
\end{aligned} \tag{4}$$

The Rademacher penalty, as computed in Eq. (4), provides a measure of complexity for a learner's hypothesis space, by determining how well it will satisfy, through learning, two very dissimilar training sets. Note that a learner, which attempts to achieve zero training error, on both h_1 and h_2 simultaneously, will produce a large Rademacher penalty, since it will attempt to satisfy two such dissimilar training sets exactly. In the empirical results in Section 6, the Rademacher complexity penalty will be computed for all participating neural networks to determine how complex they can become with a noisy learning problem.

2.3. Combining classifiers with boosting

Boosting can be thought of as a process for incremental improvement of a learner's hypothesis (Schapire, 1990). This improvement can be achieved through construction of an aggregate hypothesis. In fact, it has been shown that some decision tree learning algorithms are indeed boosting algorithms (Kearns and Mansour, 1995). With boosting, the goal is to combine, possibly many, simple classifiers into an agglomerated classifier, or improve the performance of an existing adaptive classifier as it is exposed to different training data (Lazano, 2000; Vapnik, 1995; Vapnik, 1998). It can be shown that boosting will allow agglomerative learning algorithms to achieve a desired error tolerance while structural risk minimization can be employed to keep the combined hypothesis from becoming too complex (Vapnik, 1998).

In relation to boosting, Fuzzy ART performs aggregate accumulation of F_2 nodes while Fuzzy ARTMAP controls how these are formed in an effort to improve its performance on the training data. Composition of the aggregate hypothesis in Fuzzy ARTMAP can continue without bound, which allows its complexity to grow without bound. What is needed is a methodology for determining when the error in Fuzzy ARTMAP is large enough to necessitate aggregate growth, but not at the expense of hypothesis complexity in terms of overall generalization error. The search for this new methodology motivated the research presented in this paper.

3. Fuzzy ART and Fuzzy ARTMAP

In this section a brief review of Fuzzy ART and Fuzzy ARTMAP is provided to make it easier to understand the modifications proposed later in this paper. Fuzzy ARTMAP is a neural network architecture designed to learn a mapping between example instances and their associated labels (Carpenter et al., 1992). Fuzzy ARTMAP is composed of

two Fuzzy ART neural network modules connected through a MAP field, as shown in Fig. 3. The mapping formed by Fuzzy ARTMAP actually consists of two separate mappings in composition. The first mapping occurs in the Fuzzy ART modules where data is clustered into categories, and thus each data sample, presented to the *A*-side Fuzzy ART module (see Fig. 3), maps to a single cluster template. Then each *A*-side Fuzzy ART cluster template is mapped to a single *B*-side Fuzzy ART cluster template, representing a data label, through the Fuzzy ARTMAP MAP field. The overall mapping learned by Fuzzy ARTMAP is a composition of these two separate mappings.

3.1. Fuzzy ART

Carpenter et al. (1991) provides a complete description of Fuzzy ART. The research in this section will focus on understanding and enhancing the vigilance criterion and how it is used to determine clusters in the F_2 node templates. The best matching F_2 node from the choice competition, J, must satisfy the vigilance criterion

$$\frac{|\mathbf{I} \wedge \mathbf{w}_J|}{|\mathbf{I}|} \geq \rho. \tag{5}$$

The vigilance parameter, ρ in Eq. (5), is a user-supplied input in the interval (0, 1). Note that at least one F_2 node, the uncommitted node, will always satisfy the vigilance criterion. The maximum choice F_2 template node satisfying the vigilance criterion is allowed to learn the input vector, a condition called *resonance*.

3.1.1. Fuzzy ART F_2 node category template

When using fast learning, a committed Fuzzy ART F_2 node j has a weight vector defined as $\mathbf{w}_j = \mathbf{x}_1 \wedge \mathbf{x}_2 \wedge \ldots \wedge \mathbf{x}_n$, where F_2 node j has learned all of the input data points in $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, and $\wedge$ denotes the logical AND operation. Because of teh complement codingused in this network, $\mathbf{w}_j$ defines the minimum hyperbox containing the data points in X. The vigilance criterion ensures that

$$|\mathbf{w}_j| = \sum_{i=1}^{2m} w_{ji} = \sum_{i=1}^{2m} \min_{k=1}^{n} x_{ki} \geq \rho. \tag{6}$$

Thus, $\mathbf{w}_j = (\mathbf{p}, \mathbf{q}^c)$ where $p_k = \min_{i \in \{1,2,\ldots,n\}} x_{ik}$ and $q_k = \max_{i \in \{1,2,\ldots,n\}} x_{ik}$. The axis-parallel hyper-rectangle for $\mathbf{w}_j$ has a minimum point at $\mathbf{p}$ and a maximum point at $\mathbf{q}$. The first m points from $\mathbf{w}_j$ are the 'lower left' corner, and the second m points are the complement of the 'upper right' corner of the hyperbox defined by the F_2 node j. The vigilance parameter, ρ, can be used to control the granularity of clusters covering the problem space. A larger ρ value will force Fuzzy ART to create smaller clusters, necessitating more clusters to cover a larger problem space. A smaller ρ value will allow Fuzzy ART to create larger clusters, meaning fewer clusters are needed to cover a problem space.

The region of attraction of a Fuzzy ART F_2 node j contains all of the data points within the vigilance boundary of the hyperbox defined by $\mathbf{w}_j$. In Fig. 7(a), the region of attraction is shown for a two-dimensional Fuzzy ART F_2 node with weights $\mathbf{w}_j = (0.1, 0.6, 0.8, 0.3)$ where the vigilance is 0. As the value of the vigilance parameter is increased, there is not much change in the region of attraction for $\mathbf{w}_j$ until $\rho = 0.5$ as shown in Fig. 7(b). Further increasing the value of the vigilance parameter, the region of attraction for $\mathbf{w}_j$ shrinks and approaches the size of the hyperbox defined by $\mathbf{w}_j$ seen in Fig. 7(c)–(e). At a vigilance value of 0.9 the region of attraction of w_j becomes precisely the hyperbox itself as seen in Fig. 7(e). At vigilance values greater than 0.9, w_j cannot attract any input points at all, including the ones inside the hyperbox it defines ($|\mathbf{w}_j| < \rho$ implies $|\mathbf{w}_j \wedge \mathbf{I}|$ for all $\mathbf{I} = (\mathbf{x}\mathbf{x}^c)$ when $\rho > 0.9$).

3.2. Fuzzy ARTMAP

Carpenter et al. (1992) provide a complete description of Fuzzy ARTMAP. The research described in this section focuses on analyzing and enhancing the operation of the Fuzzy ARTMAP MAP field. The Fuzzy ARTMAP MAP field, shown in Fig. 3, links data cluster templates (*A*-side) with label cluster templates (*B*-side). Supervised learning is performed in Fuzzy ARTMAP by ensuring that each *A*-side template is linked with only one *B*-side template. Thus, a many-to-one association from data pattern templates to label templates is formed in the Fuzzy ARTMAP MAP field.

The Fuzzy ARTMAP MAP field weights, w_{jk}^{AB}, are used to control associations between *A*-side F_2 nodes and *B*-side F_2 nodes. An uncommitted *A*-side F_2 node, j, has the following initial weight values

$$w_{jk}^{AB} = 1, \forall k, 0 \leq k \leq N^B, \tag{7}$$

meaning that j is not currently associated with any *B*-side F^2 node (there are N^B *B*-side F_2 nodes), and in fact it is available for future learning (association through the MAP field). An uncommitted *A*-side F_2 node j becomes committed with *B*-side F_2 node K through the following weight assignments

$$w_{jK}^{AB} = 1 \quad \text{and} \quad w_{jk}^{AB} = 0, \quad \forall k \neq K, \tag{8}$$

thus *A*-side F_2 node, j, is exclusively and permanently linked with *B*-side F_2 node, K.

The Fuzzy ARTMAP architecture ensures the many-to-one mapping through the use of a match tracking lateral reset, as shown in Fig. 3.

3.3. Modifications to fuzzy ARTMAP

Many modifications to Fuzzy ARTMAP have been proposed in the literature. In this section, these architectures will be described in terms of how they address the weaknesses described above. To avoid the problem of over-fitting, a Fuzzy ARTMAP-based neural network must deal with conflicting training data without the need for maintaining strictly perfect training performance. One way of relaxing the training requirements of Fuzzy ARTMAP is to use a statistical approach. Several of the following techniques employ a

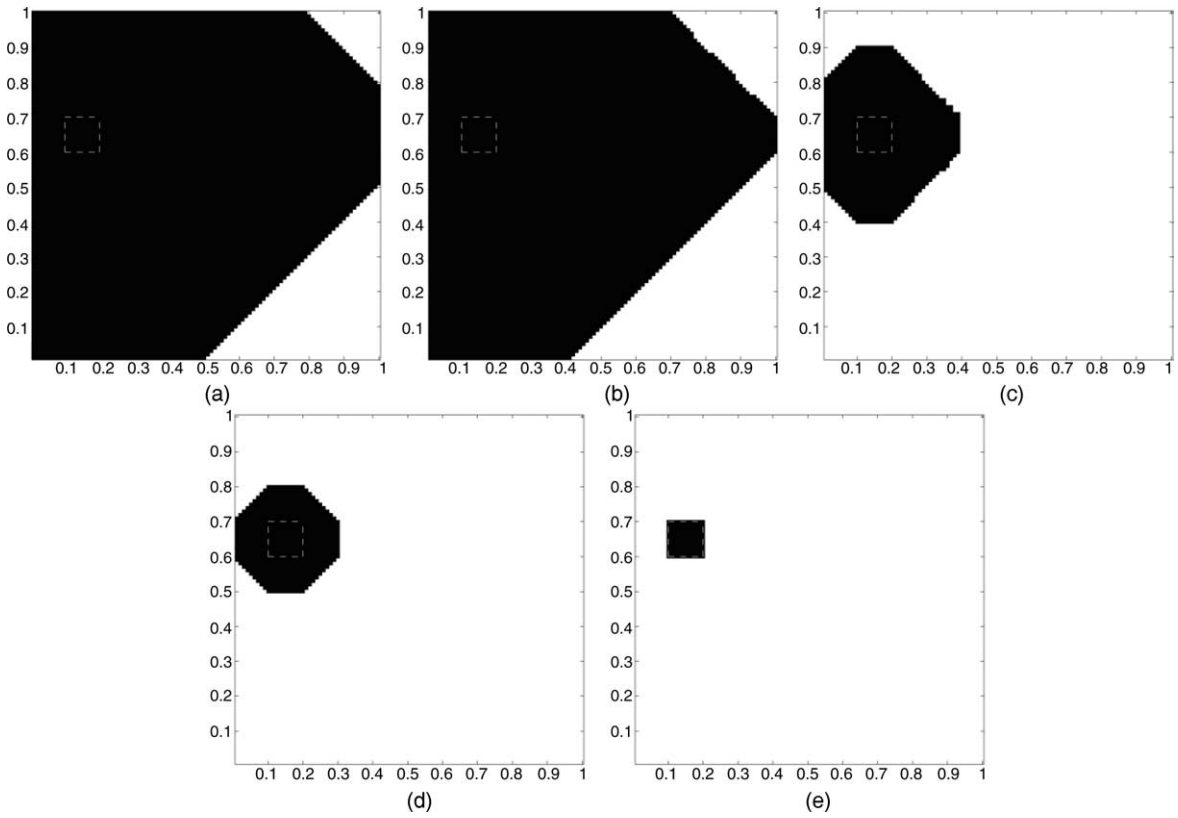

Fig. 7. Region of attraction of a single Fuzzy ART cluster template at (a) $\rho=0.0$, (b) $\rho=0.5$, (c) $\rho=0.8$, (d) $\rho=0.85$, and (e) $\rho=0.9$.

statistical learning approach. These modifications to Fuzzy ARTMAP either directly or indirectly reduce the representation error, the approximation error or a combination of both.

For instance, Gaussian ARTMAP uses F_2 nodes which have Gaussian spatial extents, differing from Fuzzy ARTMAP's hyperboxes in their representation (Williamson, 1996). As one would expect, Gaussian ARTMAP performs very well on learning problems composed of Gaussian distributions, and this will be seen in the empirical results later in this paper. However, Gaussian ARTMAP might not perform as well as Fuzzy ARTMAP on learning problems involving linear boundaries, and this will also be seen in the empirical results. Gaussian ARTMAP also relaxes the restriction of zero training error by virtue of its representation (the hyper-Gaussians have infinite extent) and by adding a maximum limit to the number of training epochs allowed as opposed to training until a stable solution is obtained. Without allowing the network to reach a steady state on its own, zero training error cannot be guaranteed in Gaussian ARTMAP.

ART-EMAP uses a previously trained Fuzzy ARTMAP network but combines information from multiple F_2 nodes during the non-learning phase to reduce errors from noisy training data (Carpenter and Ross, 1995). In the empirical results later in this paper, it will be seen that ART-EMAP does indeed improve performance where certain kinds of noise are involved, but it does not affect the learning performed or the complexity of hypotheses generated, and so it has the same effective hypothesis complexity measure as Fuzzy ARTMAP. ARTMAP-IC adds instance counting to Fuzzy ARTMAP which gives a relative weighting to each F_2 node (Carpenter and Markuzon, 1998). Those F_2 nodes which resonate with many examples will have a high instance count, and those resonating with few samples will have a low instance count. It is unclear if instance counting, by itself, will have any effect in situations of learning overlap. Distributed ARTMAP improves upon ARTMAP-IC by allowing multiple (distributed) activation of F_2 nodes during learning (Carpenter et al., 1998). Specifically, Distributed ARTMAP allows more than one of the available F_2 nodes to be active during resonance learning, and information from this ensemble of F_2 nodes is combined in the predictive phase. Because of its distributed activation of cluster nodes, it is expected that distributed ARTMAP will, in general, require fewer nodes to reach a solution, which could result in better performance. In the empirical results presented later in this paper it will be shown that distributed ARTMAP can require fewer resources than Fuzzy ARTMAP in learning, but this can result in less stable solutions and thus slightly worse overall performance.

The MAP field of Fuzzy ARTMAP was changed in PROBART to accumulate association information between A-side and B-side F_2 nodes during learning. One way classification is achieved using PROBART is by selecting the maximum associated MAP field link during prediction. This type of prediction will be used for PROBART in the empirical results section below.

In Micro ARTMAP, the MAP field of PROBART is used to accumulate A-side to B-side associations and this information is used to estimate the entropy of each F_2 node, which is then reduced during learning (Gómez-Sánchez et al., 2002). In particular, Micro ARTMAP relaxes the zero training error performance of Fuzzy ARTMAP and instead calculates entropy for each F_2 node and overall network entropy, which is then reduced during learning. It is expected that Micro ARTMAP will result in improved performance in situations of learning overlap due to its minimization of cluster node entropy and overall entropy during learning.

In Section 4, a modification to Fuzzy ARTMAP is proposed for allowing an increased margin of training error, which thereby decreases the number of F_2 layer nodes used for the purpose of increasing the overall generalization error performance, specifically on learning problems involving overlapping class distributions. The empirical results in Section 6 will provide comparison of the modifications proposed in this paper with the architectures mentioned above.

4. Boosted ART and boosted ARTMAP

The research described in this paper focuses on improving the generalization error performance of Fuzzy ARTMAP, and Fuzzy ART-based architectures, particularly in situations where there is significant overlap between classes due to noise or other causes. The focus of this section involves a modification to Fuzzy ART as well as a modification to Fuzzy ARTMAP. The modification to Fuzzy ART is called Boosted ART wherein the vigilance criterion is now applied independently to each F_2 node. The modification to Fuzzy ARTMAP is called Boosted ARTMAP and consists of allowing multiple simultaneous associations in the MAP field similar to PROBART, with the addition of an error tolerance parameter to allow controlled error in the MAP field. It is important to note that Boosted ART is a generalization of Fuzzy ART and Boosted ARTMAP is a generalization of both Fuzzy ARTMAP and PROBART.

4.1. Boosted ART

The Boosted ART neural network architecture is shown in Fig. 8. In this modified architecture, each F_2 layer node now has its own vigilance parameter. The behavior of a specific F_2 node can be controlled independently using its associated vigilance parameter. Note that there is still a baseline vigilance parameter, ρ, for the entire Boosted ART module, and it is used when a node is first committed (i.e. when F_2 node $\mathbf{y}_j$ is first committed, $\rho_j = \rho$). The uncommitted node always has its vigilance set to the baseline value. The vigilance criterion for F_2 node j becomes

$$\frac{|\mathbf{I} \wedge \mathbf{w}_j|}{|\mathbf{I}|} \geq \rho_j. \tag{9}$$

Eq. (9) differs from the Fuzzy ART vigilance test, shown in Eq. (5), in that each F_2 node now has its own associated spatial extent parameter, ρ_j. The choice function for the Boosted ART module becomes

$$J = \arg\max_{0 \leq j \leq N} T_j(\mathbf{I}), \tag{10}$$

where

$$T_j(\mathbf{I}) = \begin{cases} \dfrac{|\mathbf{I} \wedge \mathbf{w}_j|}{\alpha + |\mathbf{w}_j|}, & \text{if } \dfrac{|\mathbf{I} \wedge \mathbf{w}_j|}{|\mathbf{I}|} \geq \rho_j \\ 0, & \text{otherwise,} \end{cases} \tag{11}$$

which is only slightly different from Fuzzy ART's choice function. The behavior of the Boosted ART module is similar to the Fuzzy ART module, except for the new vigilance criterion in Eq. (9). Note that the behavior of Boosted ART will reduce exactly to that of Fuzzy ART when all F_2 node vigilance values are maintained at the same value as the baseline vigilance at all times, and thus Eqs. (9) and (10) reduce to their associated Fuzzy ART counterparts.

In order to better understand the representation capabilities of Boosted ART, consider the spatial extents of two F_2 nodes in the Fuzzy ART architecture versus the Boosted ART

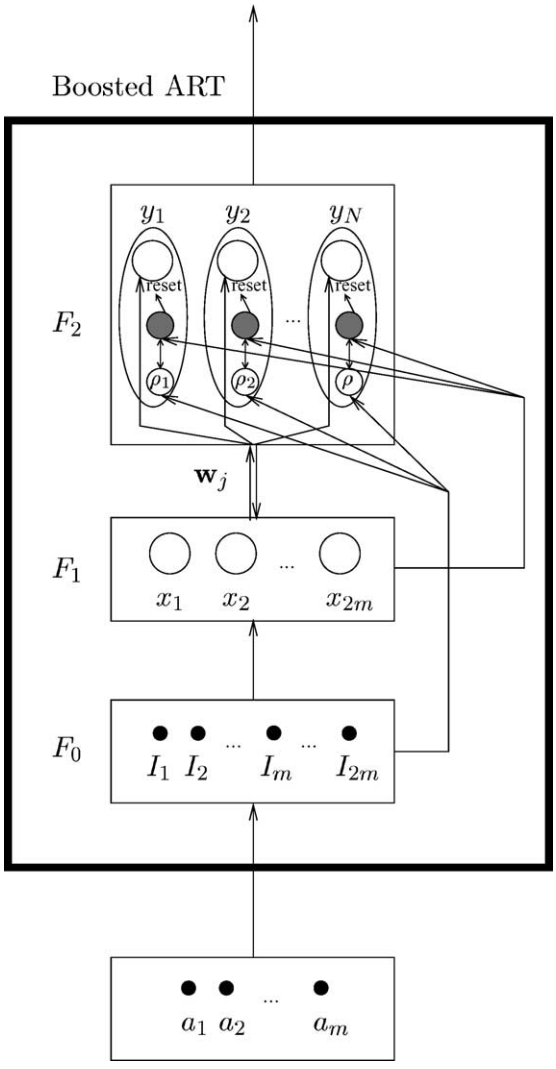

Fig. 8. The Boosted ART architecture.

architecture. In Fig. 2(b), the spatial extents of two F_2 nodes in Fuzzy ART at a vigilance of $\rho = 0.7$ are shown. Because, Fuzzy ART has only the single vigilance parameter, these two F_2 nodes cannot be used to exactly represent the two hyperboxes, shown as white-dashed lines in Fig. 2(b). In Boosted ART, however, these two F_2 nodes can each have their own vigilance values, and thus, the network is capable of representing the hyperboxes precisely, as shown in Fig. 4(a). It should be noted, however, that Fuzzy ART can represent these two spatial regions as well, but at the cost of more F_2 nodes. Specifically, Fig. 4(b) shows that ten Fuzzy ART F_2 nodes can precisely represent the two square regions at a vigilance value of $\rho = 0.9$. In general, Fuzzy ART can require exponentially more F_2 nodes to represent the same domain space as Boosted ART, where the exponent is the dimension of the domain (i.e. the number of features).

4.2. Boosted ARTMAP and the boosted ARTMAP MAP field

Similar to Fuzzy ARTMAP, Boosted ARTMAP is composed of two ART modules, except that in this case they are connected by the Boosted ARTMAP MAP field. The Boosted ARTMAP MAP field is an extension of PROBART (Marriott and Harrison, 1995), which is itself a modification of Fuzzy ARTMAP. In PROBART, the information gathered in the MAP field is not used for error reduction directly during learning. In Boosted ARTMAP, the information gathered in the MAP field is used to estimate the error in association between A-side and B-side F_2 nodes, and this error is regulated during learning.

In Boosted ARTMAP shown in Fig. 9, each A-side ART template can be associated with many, even all, B-side ART templates. The label predicted for a specific A-side ART template is the B-side ART template with the greatest MAP field frequency association value. Given a random example, $(\mathbf{x}, \mathbf{y})$, the estimate of error in a Boosted ARTMAP F_2 node will be defined as the product of the estimate of the probability that x chooses A-side ART F_2 template j times the estimate of the probability that $\mathbf{y}$ is not the predicted label associated with A-side ART template j through the MAP field. The error is estimated using the frequency information maintained in the MAP field weights, $\mathbf{w}^{AB}$, and the total number of samples seen so far, denoted S^*. Thus, given a trained Boosted ARTMAP architecture, the estimate of the total error in the MAP field is computed as

$$e_{\text{Total}} = \sum_{j=1}^{N^A-1} p_j e_j = \frac{\sum_{j=1}^{N^A-1} \left(\left| \mathbf{w}_j^{AB} \right| - \max_k \left\{ w_{jk}^{AB} \right\} \right)}{S^*}. \tag{12}$$

where

$$p_j = \Pr\left\{ \mathbf{x} \text{ chooses } v_j^A \right\} = \frac{\left| \mathbf{w}_j^{AB} \right|}{S^*} \tag{13}$$

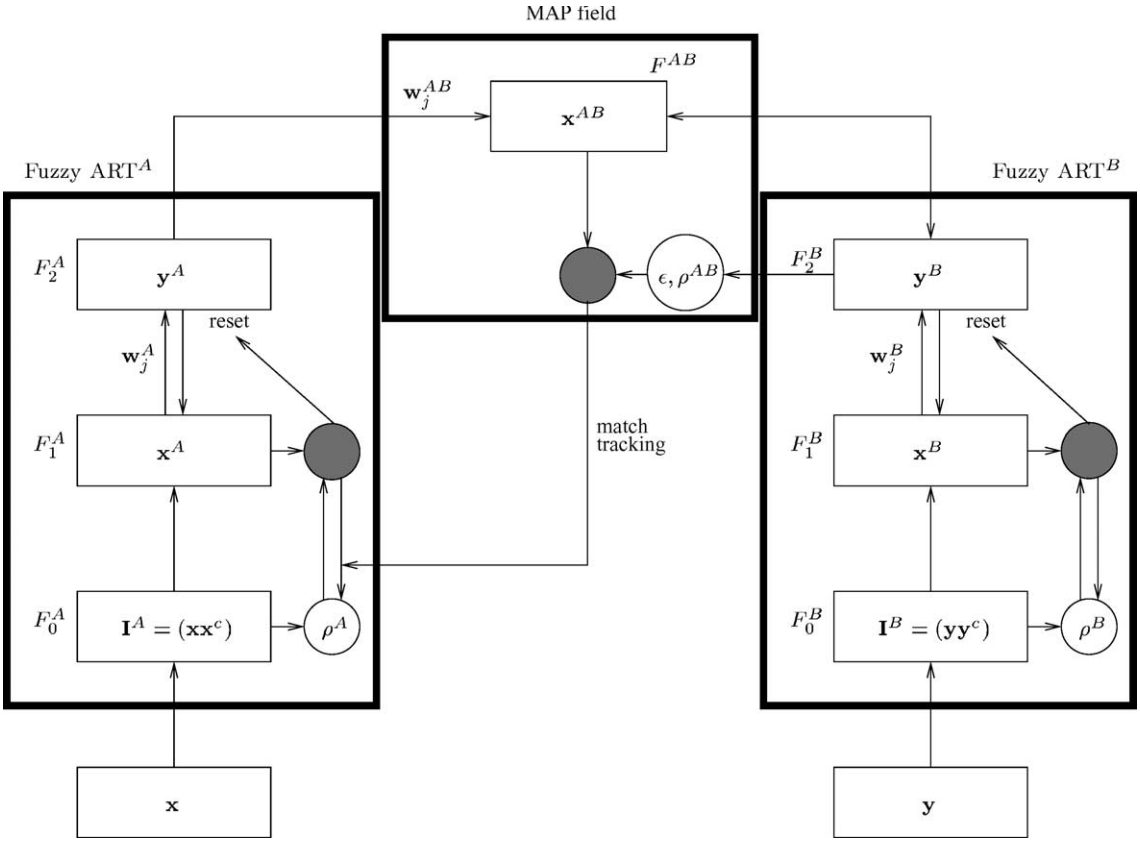

Fig. 9. The Boosted ARTMAP (BARTMAP) architecture.

and

$$e_j = \Pr\{\mathbf{y} \text{ is not predicted by } v_j^A\} = 1 - \frac{\max_k \{w_{jk}^{AB}\}}{\left|\mathbf{w}_j^{AB}\right|} \tag{14}$$

with $k = 1,\ldots,N^B - 1$ and v_j^A is a A-side Fuzzy ART template. Note that S^* can be easily computed from the MAP field weights, $S^* = \left|\mathbf{w}_j^{AB}\right|$. The error estimate of a particular A-side Fuzzy ART template, j, is $p_j \cdot e_j$.

4.2.1. Boosted ARTMAP learning

This section will provide a description of how learning is accomplished in Boosted ARTMAP. Boosted ARTMAP allows F_2 nodes from the A-side ART module to simultaneously associate with all F_2 nodes in the B-side. The association frequencies between A-side and B-side nodes are maintained in the MAP field. An uncommitted A-side F_2 node j has the following initial weight values

$$w_{jk}^{AB} = 0, 1 \le k \le N_C, \tag{15}$$

where N_C is the number of classes. Note that for the results in this paper, N_C will also be the number of nodes in the B-side Fuzzy ART module. During learning, when j is the maximum choice A-side F_2 node and k is the maximum choice B-side F_2 node, the specific weight between j and k is updated as follows

$$w_{jk}^{AB} = w_{jk}^{AB} + 1. \tag{16}$$

Thus association frequencies are computed in Boosted ARTMAP, and this information is used to bind the training error for each A-side F_2 node.

The estimate for the performance error of a committed node, j, in Boosted ARTMAP is shown in Eq. (14). In order to bind the training error during learning, the frequency information gathered in the MAP field is used in conjunction with the lateral reset match tracking mechanism of Fuzzy ARTMAP. The input error tolerance parameter, $\varepsilon \in [0, 1.0]$, is used along with ρ^{AB} to control the lateral reset. The Boosted ARTMAP MAP field lateral reset test is

$$\begin{aligned} &(1 - e'_J)|\mathbf{y}^B \wedge w'_J| < (1-\varepsilon)\rho^{AB}|\mathbf{y}^B| \Rightarrow \text{lateral reset}, \\ &(1 - e'_J)|\mathbf{y}^B \wedge w'_J| \ge (1-\varepsilon)\rho^{AB}|\mathbf{y}^B| \Rightarrow \text{resonance}, \end{aligned} \tag{17}$$

where e'_J is the interim estimated error, and $\mathbf{w}'J$ is the interim template pattern weight vector, should F_2 node J be allowed to learn the current training sample. The interim set of weights, w'_J is determined as

$$w'_{jk} = \begin{cases} 1, & \text{if } k = \text{argmax}_{1 \le k \le N_C} w_{jk}^{AB} \\ \lceil 0 + \varepsilon \rceil, & \text{otherwise.} \end{cases} \tag{18}$$

Notice that for $\varepsilon > 0$, $w'_{jk} = 1 \,\forall\, k$. The values for w'_{jk}, here, allow each A-side F_2 node to associate with any or all B-side F_2 nodes given that the estimated node error for J is below ε. The interim estimated F_2 node error value is determined as

$$e'_j = 1 - \frac{\max_{1 \le k \le N_C} w''_{jk}}{\sum_{k=1}^{N_C} w''_{jk}}. \tag{19}$$

where

$$w''_{JK} = w_{JK}^{AB} + 1 \qquad w''_{Jk} = w_{Jk}^{AB}, \quad \forall k \ne K \tag{20}$$

For an uncommitted A-side F_2 node j, the MAP field weights are

$$w_{jk}^{AB} = 0 \Rightarrow w'_{jk} = 1, \quad \forall k, \tag{21}$$

with $e'_j = 0$, and thus no lateral reset will occur since

$$(1 - e'_j)|\mathbf{y}^B \wedge w'_j| = |\mathbf{y}_B| = 1 \ge (1-\varepsilon) = (1-\varepsilon)\rho^{AB}|\mathbf{y}^B|. \tag{22}$$

In the situation where a committed A-side F_2 node is chosen, consider training sample (x, y) presented to the Boosted ARTMAP architecture, where J is the chosen A-side F_2 node, and K is the chosen B-side F_2 node. If increasing w_{JK}^{AB} by one would increase J's estimated error performance, Eq. (19) from Eq. (14), to a value greater than ε, a lateral reset occurs since

$$\begin{aligned} (1 - e'_j)|\mathbf{y}^B \wedge \mathrm{w}'_j| &= (1 - e'_j)|\mathbf{y}^B| = (1 - e'_j) < (1-\varepsilon) \\ &= (1-\varepsilon)\rho^{AB}|\mathbf{y}^B|. \end{aligned}$$

Note that $(1 - e'_J) < (1-\varepsilon) \Rightarrow \varepsilon < e'_J$, or in other words, the interim estimated error of A-side F_2 node j is greater than ε.

In Boosted ARTMAP, the application of the lateral reset is precisely as in Fuzzy ARTMAP. In fact, the performance of Boosted ARTMAP is exactly the same as Fuzzy ARTMAP, except for the use of frequency estimation during lateral reset of the MAP field.

4.2.2. Reduction of boosted ARTMAP to fuzzy ARTMAP

A major advantage behind the design of Boosted ARTMAP is that it reduces in functionality to Fuzzy ARTMAP when the desired error tolerance is set to zero. Consider the following theorem.

Theorem 4.1. Given $\rho^{AB} > 0.5$ and $\varepsilon = 0$, Boosted ARTMAP's MAP field reduces to Fuzzy ARTMAP's MAP field.

Proof. This statement will be proved by induction. First, establish a base case for both an uncommitted A-side F_2 node and a committed A-side F_2 node which has learned a single pattern.

For the case of the uncommitted node with input sample $(\mathbf{x}, \mathbf{y})$, J is the maximum choice A-side F_2 node, K is the maximum choice B-side F_2 node, and J is an uncommitted node. Now $w_{Jk}^{AB} = 0 \;\forall\, k$, $e_J = 0$, $w'_{Jk} = 1 \;\forall\, k$, and $e'_J = 0$. Therefore the Boosted ARTMAP MAP field lateral reset test, Eq. (17), reduces to

$$\begin{aligned} &|\mathbf{y}^B \wedge \mathrm{w}'_J| < \rho^{AB}|\mathbf{y}^B| \Rightarrow \text{lateral reset} \\ &|\mathbf{y}^B \wedge \mathrm{w}'_J| \ge \rho^{AB}|\mathbf{y}^B| \Rightarrow \text{resonance}. \end{aligned} \tag{23}$$

Note that this is precisely the same as used in Fuzzy ARTMAP since w'_J has the same value as $\mathbf{w}_J^{AB}$ in Fuzzy ARTMAP.

For the case where J is a committed node that has learned a single sample, assume that it has already been associated with B-side F_2 node K', then $w^{AB}_{JK'} = 1$ and $w^{AB}_{Jk} = 0 \ \forall\, k \neq K'$, $e_J = 0$, and $w'_{JK'} = 1$ and $w'_{Jk} = 0 \ \forall\, k \neq K$. For a new sample $(\mathbf{x}, \mathbf{y})$, J is the maximum choice A-side F_2 node and K is the maximum choice B-side F_2 node. If $K = K'$, then e'_J would be zero, but if $K \neq K'$, then e'_J would be 0.5. Given that $\rho^{AB} > 0.5$, J can only resonate with K if $K = K'$, otherwise a lateral reset is produced. Again, this is precisely how the Fuzzy ARTMAP MAP field operates.

For the inductive step, assume, without loss of generality, that A-side F_2 node J has learned $(n-1)$ examples all associated with the same B-side F_2 node K'. For a new sample $(\mathbf{x}, \mathbf{y})$, J is the maximum choice A-side F_2 node, and K is the maximum choice B-side F_2 node. Also, $w^{AB}_{JK'} = (n-1)$, $w^{AB}_{Jk} = 0 \ \forall\, k \neq K'$, $e_J = 0$, $w'_{JK'} = 1$ and $w'_{Jk} = 0 \ \forall\, k \neq K'$. In this case the lateral reset test will be

$$\begin{aligned} &|\mathbf{y}^B \wedge \mathrm{w}'_J| < \rho^{AB}|\mathbf{y}^B| \Rightarrow \text{lateral reset} \\ &|\mathbf{y}^B \wedge \mathrm{w}'_J| \geq \rho^{AB}|\mathbf{y}^B| \Rightarrow \text{resonance.} \end{aligned} \tag{24}$$

Note that, again, this test is precisely the same as the Fuzzy ARTMAP MAP field, since w'_J attains the same values as $\mathbf{w}^{AB}_J$ does for Fuzzy ARTMAP. □

It is interesting to note that Boosted ARTMAP not only reduces to Fuzzy ARTMAP, as shown in the previous proof, but it also reduces to PROBART when $\varepsilon = 1.0$. For this value of the error tolerance parameter, Boosted ARTMAP operates precisely as PROBART, with maximum frequency prediction. PROBART can be operated with other prediction mechanisms, but Boosted ARTMAP is designed to use a maximum frequency prediction.

The Boosted ARTMAP neural network architecture has a couple of distinct advantages. First, the training error of a Boosted ARTMAP network is explicitly bounded by the desired error tolerance parameter, ε. Each F_2 node in the A-side Fuzzy ART module of a Boosted ARTMAP network is forced to have a training error no greater than ε, which maintains an overall training error below ε. If ε is set to zero, then Boosted ARTMAP reduces exactly to Fuzzy ARTMAP in training and testing performance. Boosted ARTMAP uses on-line learning similar to Fuzzy ARTMAP. Finding the best value for ε on a particular learning problem is not easy, however, a good value for ε can be achieved by performing a simple binary search between 0.0 and 0.5. This method for determining ε works best when the error surface is fairly smooth, but a general idea of the effect ε on many learning problems can be found by checking $\varepsilon \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$.

As an example of learning in Boosted ARTMAP, consider the behavior of Boosted ARTMAP on a simple learning problem

$$\begin{aligned} &\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7, \mathbf{x}_8\} \\ &= \{(0.0,0),(0.75,1),(0.1,0),(0.2,0),(0.3,0),(0.7,1),(0.8,1),(1.0,0)\} \end{aligned} \tag{25}$$

from Dagher (1997). Training Boosted ARTMAP with $\varepsilon < 0.5$ using the first seven samples is identical to Fuzzy ARTMAP, $\mathbf{w}^A_1 = (0.00, 0.70)$, $\mathbf{w}^A_2 = (0.70, 0.20)$, $\mathbf{w}^B_1 = (0)$ and $\mathbf{w}^B_2 = (1)$. Here, there are two B-side F_2 nodes, one for each label. There are also two A-side F_2 nodes, the first chosen by samples one and three through five, and the second chosen by samples two, six and seven. The Boosted ARTMAP MAP field has the following values $\mathbf{w}^{AB}_1 = (4, 0)$ and $\mathbf{w}^{AB}_2 = (0, 3)$. These MAP field weight values indicate that each A-side F_2 node is currently only linked with a single B-side F_2 node, and each has zero performance error estimate.

Finally, training with the last sample is dependent upon the desired error tolerance. If $\varepsilon < 0.25$ then Boosted ARTMAP will produce the same network as Fuzzy ARTMAP. However, if $\varepsilon \geq 0.25$, then Boosted ARTMAP will choose $\mathbf{w}^A_2$ for $\mathbf{x}_8$. Since the anticipated performance error estimate of $\mathbf{w}^A_2$ will be $e_2 = 1 - \frac{3}{4} = 0.25 \leq \varepsilon$, then $\mathbf{w}^A_2$ will be allowed to learn $\mathbf{x}_8$ giving $\mathbf{w}^A_1 = (0.00, 0.70)$, $\mathbf{w}^A_2 = (0.70, 0.00)$, $\mathbf{w}^B_1 = (0)$ and $\mathbf{w}^B_2 = (1)$. The MAP field weights will be $\mathbf{w}^{AB}_1 = (4, 0)$ and $\mathbf{w}^{AB}_2 = (1, 3)$. Note that these weights allow a many-to-many association between A-side and B-side F_2 nodes. Also, the total estimated performance error for the entire network is 0.125. If sample $\mathbf{x}_8$ is an outlier or its label is noisy, then Boosted ARTMAP has the capacity to deal with it without needing any new F_2 nodes, where Fuzzy ARTMAP must allocate a new F_2 node to maintain its zero training error tolerance.

5. Structural boosted ARTMAP

An idealized fuzzy ARTMAP-based learning algorithm would use structural risk minimization with on-line learning.

The Rademacher complexity penalty is computed using all of the samples seen so far, and it is more accurate when 'enough' samples have been seen. With on-line learning it is difficult to decide when enough samples have been seen, whereas with off-line learning all samples can be considered at once. In this section, a neural network architecture which employs structural risk minimization and Rademacher penalization using Boosted ART and Boosted ARTMAP is presented as a first step toward incorporating structural risk minimization into Fuzzy ARTMAP-based neural networks.

The new architecture, called Structural Boosted ARTMAP (BARTMAP-SRM), is a very simple modification of Boosted ARTMAP. The BARTMAP-SRM architecture is composed of a Boosted ART module on the A-side which accepts data input and a Fuzzy ART module on the B-side to accept associated labels. This structure is similar to Fuzzy ARTMAP and Boosted ARTMAP, shown in Figs. 3 and 9, respectively. The MAP field of BARTMAP-SRM will be the same as Boosted ARTMAP, described in Section 4.2. Instead of having an uncommitted node in the A-side Boosted ART module, BARTMAP-SRM will have one F_2 node, w^A_0, that covers the entire (complement-coded) domain, i.e. $\mathrm{w}^A_0 = (0^m, 0^m)$ for an m-dimensional domain. Note that the operation of w^A_0, will be similar to the uncommitted node of Fuzzy ART in that it acts as a catch-all for any data points that do not resonate with any of the other (committed) F_2 nodes. All F_2 nodes, including w^A_0, are

maintained such that their estimated training error is no greater than the user specified error tolerance, ε, similar to Boosted ARTMAP.

BARTMAP-SRM can be operated with on-line or off-line learning, but it will employ full structural risk minimization only with off-line learning. Rademacher penalization is computed and used only when BARTMAP-SRM is trained with off-line learning. When using on-line learning, BARTMAP-SRM will continue to add new F_2 nodes as necessary without bounds, similar to Fuzzy ARTMAP and Boosted ARTMAP. In order to achieve zero error tolerance, it might be necessary for on-line BARTMAP-SRM to assign each training sample to its own F_2 node. Note that the error tolerance is only used with on-line learning. In this mode of operation, BARTMAP-SRM demonstrates a proof of concept for the universal function approximation capabilities of Fuzzy ART-based and Fuzzy ARTMAP-based neural networks (Verzi, 2003; Verzi et al., 2003). Learning in BARTMAP-SRM is motivated by the properties of hyperboxes in Ω described next.

5.1. Axis parallel hyper-squares and open sets in $\Re^m$.

An interesting property of open sets in $\Re^m$ is that each such nonempty open set is composed of a countable union of disjoint boxes belonging to Ω, defined as

$$\mathcal{Q} = \mathcal{Q}_1 \cup \mathcal{Q}_2 \cup \mathcal{Q}_3 \cup \ldots, \tag{26}$$

where

$$\mathcal{Q}_n = \{[\mathbf{p}, \mathbf{q}) : \mathbf{p}, \mathbf{q} \in P_n \text{ and } q_i - p_i = 2^{-n}\}, \tag{27}$$

and

$$P_n = \{\mathbf{z}/2^n : z_i \in \mathcal{Z}\}. \tag{28}$$

Ω_n is the collection of all half-open boxes with sides of length 2^{-n} and corners at P_n (Rudin, 1974). This property means that open sets in $\Re^m$ can be covered by a collection of sets from Ω. It may take an infinite number of these boxes to cover a specific set, but these will be countably infinite. It is also interesting to note that the measurable sets in $\Re^m$ can be approximated by unions of sets from Ω using a fine Vitali or Vitali-like covering (DiBenedetto, 2001). In Fig. 10(a), a collection of boxes from Ω_1, Ω_2, and Ω_3 in $\Re^2$ is shown. Another interesting property of sets $\omega_1, \omega_2 \in \Omega$ is that they are either disjoint from one-another or one is a proper subset of the other, as can be seen in Fig. 10. BARTMAP-SRM uses complement coding, and thus only those boxes from Ω that lie inside the unit hyperbox will be of interest. Therefore, $\Omega^* = \{\omega \in \Omega: \omega \supset (0, 1)\}$ will be used from here on in this paper.

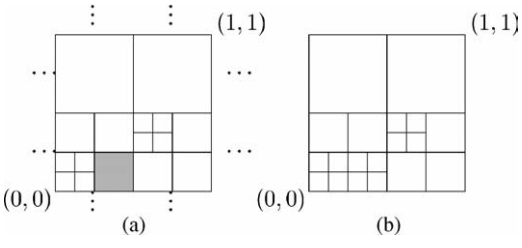

Fig. 10. Some boxes belonging to Ω for $\Re^2$.

5.2. Description of structural boosted ARTMAP

With on-line learning, BARTMAP-SRM looks very much like a hierarchical or tree-like Boosted ARTMAP. Initially BARTMAP-SRM contains only a single A-side F_2 node that covers the entire unit hyper-square. BARTMAP-SRM differs from Boosted ARTMAP when the error estimate, Eq. (19), would exceed ε. When this happens during learning, instead of a lateral reset, as would occur in Boosted ARTMAP, in BARTMAP-SRM this A-side F_2 node is split along each dimension into smaller squares at the next level in size, which are proper subsets of the F_2 node being split, as can be seen in Fig. 10. These new squares are then added to the A-side Boosted ART module. The on-line algorithm for BARTMAP-SRM is shown in Fig. 11. Resonance occurs when an A-side F_2 node is found that satisfies ε in its error estimate. As with Boosted ARTMAP, a newly added A-side F_2 node will always have an error estimate of zero. Noise in learning can be handled by increasing ε to a point where it is greater than the noise seen in the data, similar to the operation of Boosted ARTMAP. Note that in this mode of operation (on-line learning), BARTMAP-SRM does not use structural risk minimization since it uses empirical risk minimization, as do all other Fuzzy ARTMAP-based architectures mentioned in this paper.

With off-line learning, BARTMAP-SRM can perform full structural risk minimization. In this mode of operation, a series of BARTMAP-SRM networks of increasing complexity, in terms of the number of A-side F_2 nodes, will be used. The off-line BARTMAP-SRM algorithm is shown in Fig. 12. Each network at a particular level of complexity, n, is trained without growing on the entire training set. During training, the networks are not allowed to grow since the algorithm is collecting statistics about learning potential using a fixed network size. Note that the number of nodes used at a particular level of complexity is exponential in n, but this value will not be more than $m^{[\log_m(N)]}$ which is bounded by the least factor of m greater than N. The Rademacher penalty is computed as in Eq. (4) using BARTMAP-SRM at complexity n. The network with the minimum overall combination of training error plus Rademacher penalty will be output from this version of BARTMAP-SRM.

Even though, BARTMAP-SRM has the two modes of operation, only the second mode is used in the empirical results since it uses structural risk minimization with the Rademacher penalty. These results will allow a direct comparison on-line versus off-line learning and empirical risk minimization versus structural risk minimization on the learning problems.

6. Empirical results

In this section, empirical learning results are presented to demonstrate how Boosted ARTMAP (BARTMAP) can be used to take advantage of the trade-off between training error and hypothesis complexity by adjusting the error

```
\begin{figure*}
\begin{tabbing}
\=***\=***\=***\=***\=***\=***\=***\= \kill
\>  Given sample $(\mathbf{x}, \mathbf{y}), \mathbf{x} \in [0, 1]^{m}$: \\
\> 1 \> $\mathbf{w}^{A} = (\mathbf{0}^{m}, \mathbf{0}^{m})$. \\
\> 2 \> find  best matching (Fuzzy {ART}) $F^{B}_{2}$ node for $\mathbf{y}$,
call it $\mathbf{w}^{B}_{K}$. \\
\> 3 \> find  best matching (Boosted {ART}) $F^{A}_{2}$ node for
$\mathbf{x}$, call it $\mathbf{w}^{A}_{J}$. \\
\> 4 \> {\bf if} \ $\frac{w^{AB}_{JK} + 1}{|\mathbf{w}^{AB}_{J}|} >
\epsilon$ \ {\bf then} \\
\> 5 \> \> split $\mathbf{w}^{A}_{J}$ along each feature dimension and add
each \\
\>   \> \> of these new nodes to $\mathbf{w}^{A}$. \\
\> 6 \> \> goto step 3. \\
\> 7 \> update {MAP} field: $w^{AB}_{JK} = w^{AB}_{JK} + 1$. \\
\end{tabbing}
  \caption{On-line Structural Boosted {ARTMAP} Algorithm.}
  \label{sbamon}
\end{figure*}
```

Fig. 11. On-line structural boosted ARTMAP algorithm.

tolerance parameter. All architectures in these experiments use empirical risk minimization and on-line learning except for Structural Boosted ARTMAP (BARTMAP-SRM), which uses structural risk minimization and off-line learning. Thus, the empirical results will also offer a comparison of the differences between these types of learning. In general, it is expected that structural risk minimization, and BARTMAP-SRM with the off-line learning advantage, will provide better performance, especially in situations where there is noise or overlap. However, it will be interesting to see how well the Fuzzy ARTMAP-based networks with on-line learning and empirical risk minimization compare with BARTMAP-SRM. Resource usage, in terms of the number of F_2 nodes used in training, is another issue of interest here. BARTMAP-SRM is not necessarily efficient in its usage of F_2 nodes for all learning problems, but this issue is important to the learning in BARTMAP as well as Gaussian ARTMAP (Williamson, 1996), Distributed ARTMAP (Carpenter et al., 1998) and also in Micro ARTMAP (Gómez-Sánchez et al., 2000; Gómez-Sánchez et al., 2002).

The first set of results presented demonstrate the utility of BARTMAP and BARTMAP-SRM on several learning problems where noise and/or class overlap occur. A classic learning problem from Fuzzy ARTMAP will also be presented in these empirical results to show that the proposed new architectures do not significantly degrade in performance where there is no noise or overlap. Following these results, the learning performance of BARTMAP and the effect of its error tolerance parameter on generalization error and network complexity will be considered in more detail. Next, the Rademacher complexity penalty will be computed on a noisy learning problem for all Fuzzy ARTMAP-based architectures used in this section. Finally, the performance of BARTMAP-SRM will be considered in more detail.

```
\begin{figure*}
\begin{tabbing}
\=***\=***\=***\=***\=***\=***\=***\= \kill
\>  Given sample set $S = \{ (\mathbf{x}_{1}, \mathbf{y}_{1}),
(\mathbf{x}_{2}, \mathbf{y}_{2}), ..., (\mathbf{x}_{N}, \mathbf{y}_{N})i \},
\mathbf{x}_{i} \in [0, 1]^{m}$: \\
\> 1 \> {\bf for} \ $n = 1$ \ {\bf to} \ $\lceil \log_{m} (N) \rceil$ \ {\bf
do} \\
\> 2 \> \> $C = m^{n}$. \\
\> 3 \> \> construct {BARTMAP}-{SRM} network $\mathbf{w}^{A}_{n}$ with $C$
$F_{2}$ nodes \\
\> \> \>   from $\Omega^{*}_{n}$. \\
\> 4 \> \> compute training error, $L(w^{A}_{n})$ for $S$. \\
\> 5 \> \> compute Rademacher penalty, $pen(\mathbf{w}^{A}_{n}; n)$ for $S$.
\\
\> 6 \> \> $v_{n} = L(\mathbf{w}^{A}_{n}) + pen(\mathbf{w}^{A}_{n}; S)$. \\
\> 7 \> $\hat{n} = \arg \min v_{n}$. \\
\> 8 \> output network $\mathbf{w}^{A}_{\hat{n}}$. \\
  \end{tabbing}
                                2005Oct20_143438[1].txt
  \caption{Off-line Structural Boosted {ARTMAP} Algorithm.}
  \label{sbamoff}
\end{figure*}
```

Fig. 12. Off-line structural boosted ARTMAP algorithm.

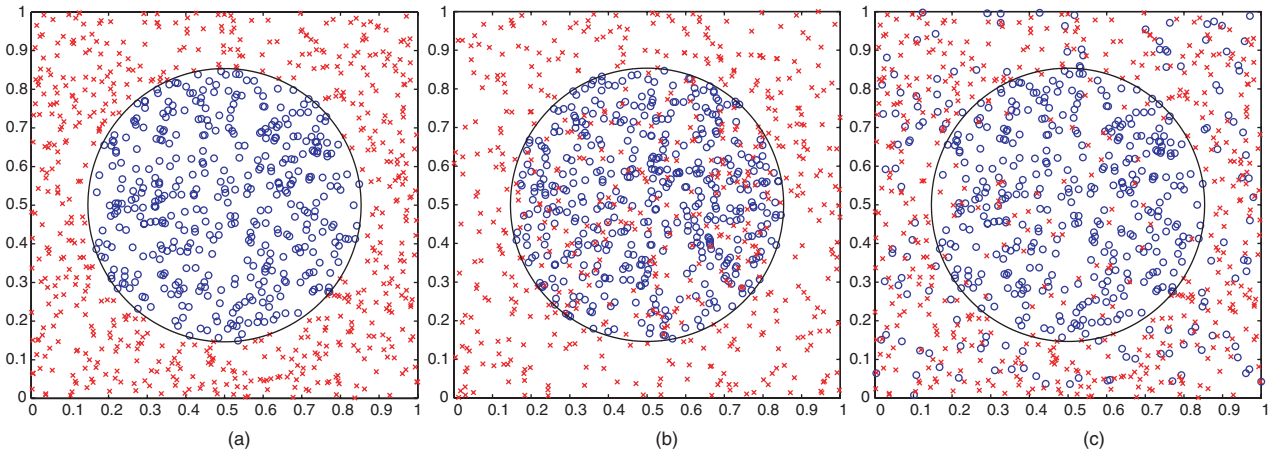

Fig. 13. Three variations on the circle-in-the-square problem, where (a) there is no overlap between the pattern classes, (b) the pattern classes overlap, and (c) the pattern classes overlap and uniform random noise has been injected.

6.1. Learning results

In this section, the generalization performance of BARTMAP and BARTMAP-SRM are compared with Fuzzy ARTMAP (FuzARTMAP) (Carpenter et al., 1992), ARTEMAP (Carpenter and Ross, 1995), ARTMAP-IC (Carpenter and Markuzon, 1998), Distributed ARTMAP (dARTMAP) (Carpenter et al., 1998), Gaussian ARTMAP (GARTMAP) (Williamson, 1996), PROBART (Marriott and Harrison, 1995), and Micro ARTMAP (μARTMAP) (Gómez-Sánchez et al., 2000; Gómez-Sánchez et al., 2002) on some statistical learning problems.

In each of the learning problems, one class was labeled 0 and the other 1. All data were normalized to fit within the unit square so that complement coding could be used. Other than the diabetes-learning problem, each class contributed equally to both the training and test data sets. For the two-dimensional generated data in these experiments, each network was trained on 1000 training samples and tested with either 1000 (in the bimodal Gaussian learning problem) or 10,000 (in the other two-dimensional generated learning problems) test samples. For each of the learning problems, 100 such training/testing scenarios were conducted to arrive at a statistical sampling of each architecture's performance. The mean and standard deviation values reported in the tables below reflect this statistical sampling.

An A-side Fuzzy ART baseline vigilance of 0.0 and B-side Fuzzy ART baseline vigilance of 1.0 was used for Fuzzy ARTMAP, and the MAP field vigilance was set to 1.0. In GARTMAP, γ values ranging from 0.01 to 0.1 were used, and GARTMAP was trained for five epochs for each learning problem. The baseline vigilance for PROBART will be specified for each problem separately. BARTMAP was trained using the same parameter values as Fuzzy ARTMAP, except the desired error tolerance values which is problem specific.

6.1.1. Circle-in-the-square problem (Carpenter et al., 1992)

In this problem, shown in Fig. 13(a), the circumference of the circle represents the optimal decision boundary. The diameter of the circular class is equal in size to the diagonal distance across a square half the size of the big square (i.e. the smaller square is circumscribed in the circle), and both are centered about the same point. The Circle-in-the-Square represents a well-separated learning problem, and it helps to demonstrate how aggregating learners, such as Fuzzy ARTMAP and Fuzzy ARTMAP-based architectures, can represent a more complex space with simple components, even when the components differ structurally than the target space. Here the goal is to learn a circle with a collection of hyperboxes. Table 1 shows the results for this learning problem. The second column shows the average number of passes through the training data, called epochs needed to reach a solution. The third column gives the average number of F_2 nodes used in training the networks. The fourth column shows the percentage of correctly classified test instances, and the last column is the standard deviation of the error percentage over the experiments conducted. Both PROBART and BARTMAP-SRM require more F_2 nodes than the other architectures. In PROBART this is due to the fact that training error is not directly controlled as it is in BARTMAP. BARTMAP-SRM needs more F_2 nodes due to the fact that this problem contains well-separated classes, which conforms better to empirical risk

Table 1
Learning results for the circle-in-the-square problem

Architecture	Epochs	F_2 nodes	% Correct	Std Dev
FuzARTMAP	7.0	24.7	95.9	0.6
ART-EMAP	7.0	24.7	88.7	4.7
ARTMAP-IC	7.0	24.7	95.9	0.6
dARTMAP	1.0	13.7	90.9	2.4
GARTMAP ($\lambda=0.1$)	5.0	11.4	85.6	16.5
PROBART ($\rho=0.85$)	2.0	67.4	89.6	1.2
μARTMAP ($h=0.15$)	49.6	17.0	93.3	3.4
BARTMAP ($\varepsilon=0$)	7.0	24.7	95.9	0.6
BARTMAP-SRM	20.0	61.0	94.0	0.6

Table 2
Learning results from the noisy circle-in-the-square problem

Architecture	Epochs	F_2 nodes	% Correct	Std Dev
FuzARTMAP	7.5	202.6	73.0	2.0
ART-EMAP	7.5	202.6	78.4	7.1
ARTMAP-IC	7.5	202.6	72.9	1.4
dARTMAP	1.0	57.8	68.0	4.8
GARTMAP ($\lambda=0.2$)	5.0	17.1	84.2	6.4
PROBART ($\rho=0.85$)	2.0	67.4	87.7	1.6
μARTMAP ($h=0.25$)	112.9	30.8	65.6	7.8
BARTMAP ($\varepsilon=0.25$)	13.3	63.8	85.3	2.1
BARTMAP-SRM	13.0	40.0	90.0	1.1

minimization. Note that BARTMAP reduces exactly to Fuzzy ARTMAP when $\varepsilon = 0$. With no noise or overlap, BARTMAP performs best using an error tolerance of 0, as shown in Table 1. It is important to note that BARTMAP has the distinct advantage over the Fuzzy ARTMAP-based architectures in that it does generalize upon Fuzzy ARTMAP and can reduce to Fuzzy ARTMAP with zero error tolerance.

6.1.2. Noisy circle-in-the-square problem.

In this problem, 20% uniform noise was added to the labels from the previous learning problem, as shown in Fig. 13(c). Thus with probability ⅕ each sample label is flipped. This label noise is significant, but it will demonstrate the performance of the learning algorithms in the presence of noise. In Table 2, μARTMAP does not handle noisy data very well as can be seen by the fact that it takes so many epochs to reach a solution. ART-EMAP does better than Fuzzy ARTMAP on this problem by combining multiple F_2 nodes for prediction. However, ART-EMAP only differs from Fuzzy ARTMAP after learning is completed, and thus both architectures have the same complexity. ART-EMAP could probably perform even better with reduced network sizes. This is precisely what BARTMAP attempts to achieve. GARTMAP shows good performance here and throughout most of the experiments in this paper, but it does have a very high standard deviation across the training sets implying it is not as stable as some of the other algorithms. PROBART does very well on this problem and approaches what can be achieved by BARTMAP-SRM. This problem shows how F_2 node proliferation can occur in Fuzzy ARTMAP in situations of noise. Here, Fuzzy ARTMAP creates many more F_2 nodes than are necessary to solve this problem. Given the label noise in this problem, BARTMAP performs best using an error tolerance of slightly larger than 0.20 so that it can create clusters which can handle at least 20% training error. The BARTMAP error tolerance used for the results in Table 2 is 0.25.

Table 3
Learning results for the overlapping circle-in-the-square problem

Architecture	Epochs	F_2 nodes	% Correct	Std Dev
FuzARTMAP	7.8	176.5	69.0	0.6
ART-EMAP	7.8	176.5	68.4	2.0
ARTMAP-IC	7.8	176.5	69.0	0.6
dARTMAP	1.0	47.6	66.6	1.9
GARTMAP ($\lambda=0.175$)	5.0	10.3	65.9	9.3
PROBART ($\rho=0.85$)	2.2	66.2	72.6	2.3
μARTMAP ($h=0.55$)	39.7	49.0	72.7	3.0
BARTMAP ($\varepsilon=0.3$)	17.6	53.3	73.6	0.9
BARTMAP-SRM	16.0	49.0	74.0	0.7

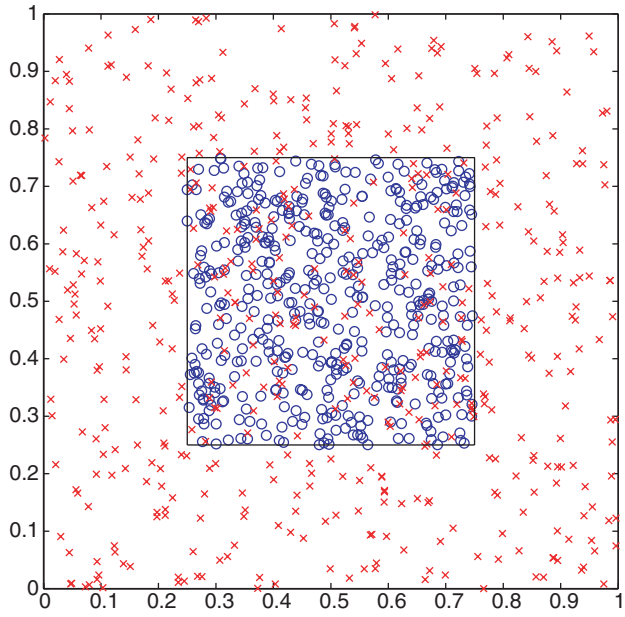

Fig. 14. The overlapping squares problem.

6.1.3. Overlapping circle-in-the-square problem.

This experiment, shown in Fig. 13(b) involves a uniformly distributed circle overlapping a uniformly distributed square. This problem is similar to the first experiment except that both data classes have samples inside the circle. Both circle and square are centered on the same point. This problem represents a case of one-sided error, where outside the circle no

Table 4
Learning results for the overlapping square problem

Architecture	Epochs	F_2 nodes	%Correct	SD
FuzARTMAP	7.7	127.6	77.9	0.7
ART-EMAP	7.7	127.6	73.4	2.4
ARTMAP-IC	7.7	127.6	77.9	0.7
dARTMAP	1.0	35.9	75.9	2.0
GARTMAP ($\lambda=0.1$)	5.0	10.8	81.9	1.7
PROBART ($\rho=0.85$)	2.1	62.5	79.7	1.2
μARTMAP ($h=0.4$)	24.4	52.7	81.2	1.9
BARTMAP ($\varepsilon=0.25$)	9.3	20.8	83.3	2.1
BARTMAP-SRM	5.0	16.0	87.5	0.3

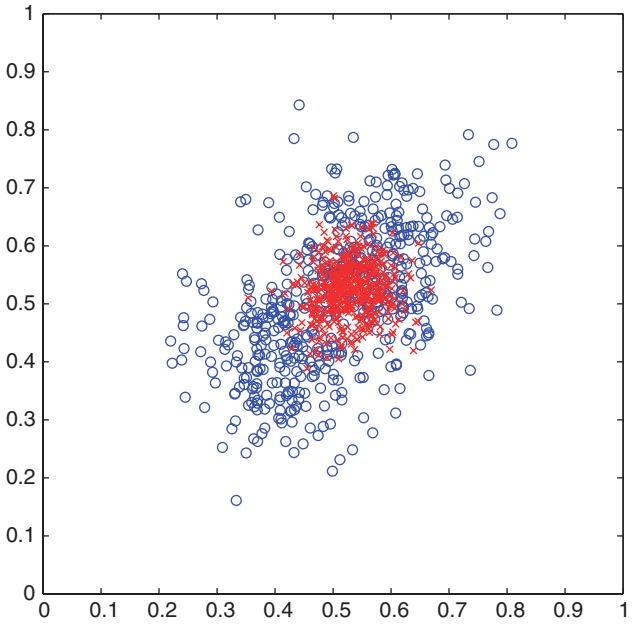

Fig. 15. The overlapping bimodal Gaussians problem.

misclassification will occur. As with the first problem, the optimal solution is the circle itself, but that sort of boundary is difficult for any of the Fuzzy ARTMAP-based architectures (except for those that use curve hyper-regions in the representation of their F_2 nodes such as Gaussian ARTMAP and Ellipsoid ARTMAP) to approximate precisely with a small number of F_2 nodes, especially given the overlap inside the circle. This particular problem is difficult for all of the architectures. It is interesting to see that ART-EMAP does not improve upon Fuzzy ARTMAP with this kind of one-sided error. It is not clear why GARTMAP has such difficulty with this kind of problem. With no noise but close to 30% overlap (of the circle onto the larger square), BARTMAP performs best with an error tolerance of 0.3, shown in the results in Table 3.

6.1.4. Overlapping squares problem.

This experiment, an example of which is shown in Fig. 14, involves a uniformly distributed square overlapping a uniformly distributed square, where the smaller square has half the area of the larger square. Both squares are centered on the same point. This problem should be easy to solve with just a small number of hyperboxes. In fact, Table 4 shows that BARTMAP-SRM achieves a nearly optimal solution with exactly 16 hyperboxes. This problem shows another case of one-sided error where the smaller square represents the optimal boundary. It is interesting that none of the architectures solves this problem with the minimum number of hyperboxes. This problem can be solved with two hyperboxes, one for the x's and one for the o's. It is surprising to see that ART-EMAP does not perform better than Fuzzy ARTMAP on this problem. Once again ART-EMAP has difficulties with one-sided error. GARTMAP does very well with this problem as opposed to the previous experiment, which is a surprise. This particular problem demonstrates the distinct advantage of off-line versus on-line learning where BARTMAP-SRM is not affected by the order of presentation of training samples, and it performs very well with a very small standard of deviation as compared with all the other architectures. The smaller square overlaps precisely 25% of the larger square, and BARTMAP performs best with an error tolerance of exactly 0.25, as shown in Table 4.

Table 5
Learning results for the overlapping bimodal Gaussians problem

Architecture	Epochs	F_2 nodes	% correct	SD
FuzARTMAP	8.4	163.4	72.2	1.7
ART-EMAP	8.4	163.4	56.5	7.0
ARTMAP-IC	8.4	163.4	72.2	1.7
dARTMAP	1.0	44.6	69.2	2.9
GARTMAP ($\lambda=0.01$)	5.0	12.5	75.5	12.2
PROBART ($\rho=0.9$)	2.1	35.6	77.2	1.8
μARTMAP ($h=0.25$)	12.0	10.6	77.9	2.3
BARTMAP ($\varepsilon=0.242$)	16.0	50.1	78.7	1.6
BARTMAP-SRM	19.3	56.0	80.5	1.5

Table 6
Learning results for the Pima Indian diabetes diagnosis problem

Architecture	Epochs	F_2 nodes	% Correct	SD
FuzARTMAP	8.7	49.4	65.8	3.6
ART-EMAP	8.7	49.4	65.0	9.9
ARTMAP-IC	8.7	49.4	65.8	3.6
dARTMAP	1.0	15.6	61.6	4.7
GARTMAP ($\lambda=0.135$)	5.0	24.0	68.5	13.9
PROBART ($\rho=0.3$)	2.0	3.0	64.7	2.2
μARTMAP ($h=0.3$)	279.6	19.5	66.1	3.9
BARTMAP ($\varepsilon=0.265$)	9.9	15.3	68.2	3.3
BARTMAP-SRM	3.9	35.0	67.8	2.7

6.1.5. Overlapping bimodal Gaussians problem (Baras and Dey, 1999).

The next experiment is a difficult problem where one bimodal 2D Gaussian sits on top of the other one, as shown in Fig. 15. The Guassian with the higher peak and lower deviation ('+') is concentrated more toward the center of the image. This problem does not have a zero training error solution. This represents a problem, which has a very different data space than Fuzzy ARTMAP-based hyperboxes. However, judicious use of the hyperboxes can cover the domain properly. The results of this experiment are shown in Table 5. This particular problem shows a case where ART-EMAP's noise reduction through multiple F_2 node activation does not succeed. As expected GARTMAP performs well on this problem, but there is a high standard of deviation in its performance. μARTMAP does very well on this problem generating very small sized networks with good performance. In this problem, it is difficult

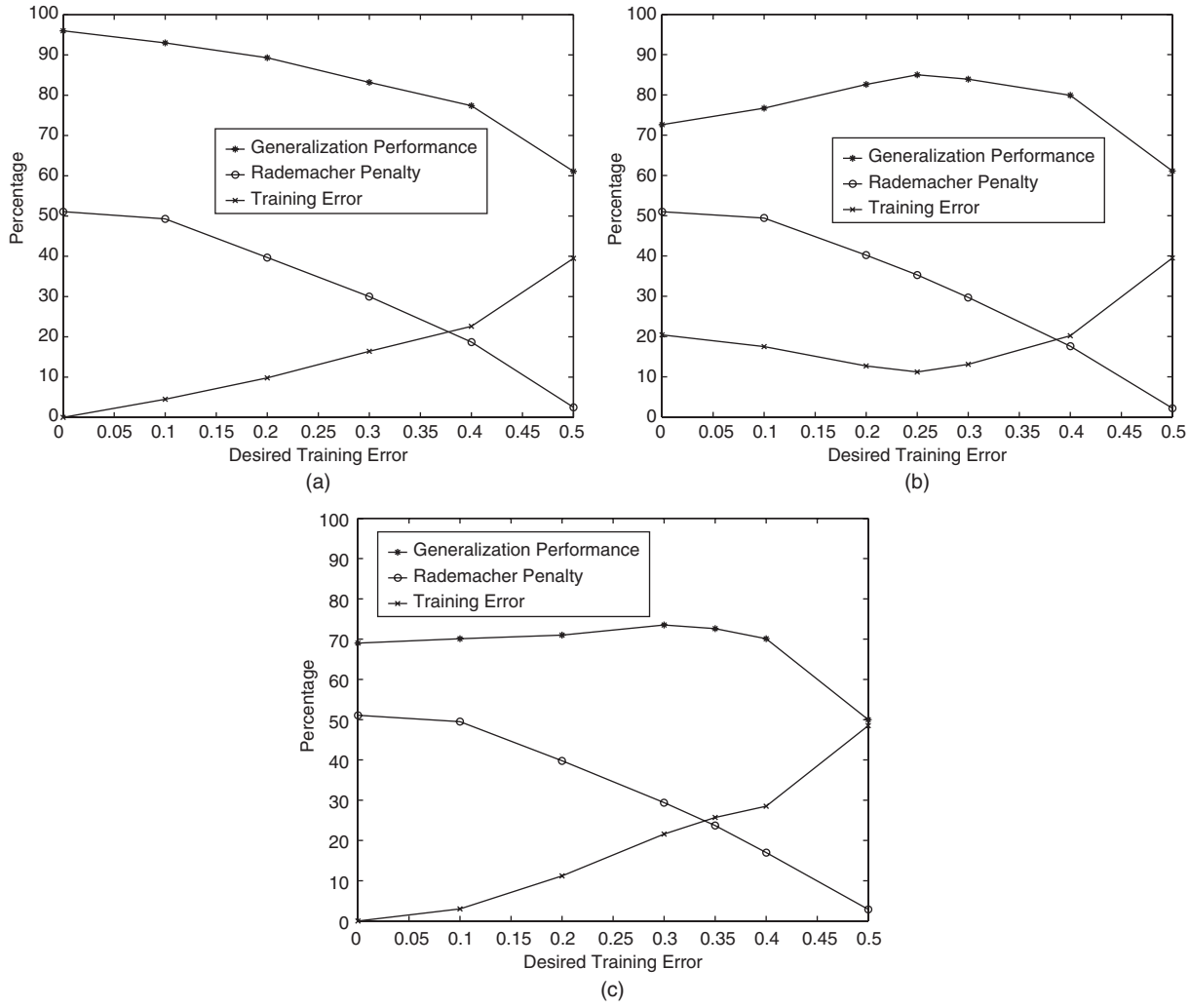

Fig. 16. Structural risk metrics for Boosted ARTMAP on the (a) circle-in-the-square, (b) noisy circle-in-the-square and (c) overlapping circle-in-the-square problems.

to characterize the amount of overlap, but after using a binary search between 0 and 0.5, a value of 0.242 was found for BARTMAP's best performance, as shown in Table 5.

6.1.6. Diabetes diagnosis problem

The next learning problem in this section comes from the Pima Indian Diabetes database in the UCI machine learning problem repository (Blake and Merz, 1998). This database consists of 768 samples (500 negative and 268 positive). These samples were split into 2/3 training data and 1/3 test data using a randomized selection without replacement for each of the 100 experiments. This example shows how all of these techniques perform on a real-world learning problem. The results of this experiment are shown in Table 6. This is a very difficult problem where none of the architectures perform very well. There is only a small amount of total samples present, and so even structural risk minimization does not do well here. In fact, both GARTMAP and BARTMAP outperform BARTMAP-SRM. This might be an indication that as a statistical off-line learning algorithm, BARTMAP-SRM requires more data to work properly, and this is expected from computational learning theory (Devroye et al., 1996; Vapnik, 1998; Vidyasagar, 1997; van der Vaart and Wellner, 1996). As in the previous learning problem, it is difficult to characterize the overlap inherent in this learning problem. Thus, a binary search between 0 and 0.5 was employed resulting in 0.265 as the best error tolerance value for BARTMAP performance, as shown in Table 6.

It is clear from the results that BARTMAP and Boosted ARTMAP-based neural network architectures, such as Structural Boosted ARTMAP offer distinct advantages in certain learning situations. BARTMAP has the added feature of reducing to Fuzzy ARTMAP for learning problems where error in training is not advantageous, by setting the error tolerance parameter to zero. Because, BARTMAP-SRM is an off-line learner, it offers a unique comparison to the other architectures, in terms of on-line versus off-line learning as well as empirical risk minimization versus structural risk minimization. Certainly in situations where there is not much training data available or where the classes being learned are well separated,

Table 7
Rademacher penalty for all architectures on the noisy circle-in-the-square problem

Architecture	Training error	Rademacher penalty
FuzARTMAP	0.0	51.3
ART-EMAP	11.3	3.7
ARTMAP-IC	0.0	51.4
dARTMAP	7.0	53.0
GARTMAP ($\lambda=0.2$)	13.4	57.0
PROBART ($\rho=0.85$)	9.4	10.7
μARTMAP ($h=0.25$)	30.9	39.2
BARTMAP ($\varepsilon=0.25$)	11.6	34.5
BARTMAP-SRM	9.3	8.4

BARTMAP-SRM can be out-performed. Even in cases where there is noise or overlap, Fuzzy ARTMAP-based architectures can come very close to the performance capabilities of BARTMAP-SRM.

6.2. Minimizing risk and boosted ARTMAP

In this section, the structural risk characteristics of Boosted ARTMAP are described for several of the learning problems previously used. The structural risk is computed using training error juxtaposed with Rademacher penalty calculated using Lazano, 2000 algorithm. The structural risk metric for the circle-in-the-square problem is shown in Fig. 16 (a). It is easy to see in Fig. 16 (a) that in trying to approximate the circle with unions of hyperboxes, a steady increase in the number of F_2 nodes produces a steady increase in generalization performance. This can be seen by traversing from right to left along the x-axis of Fig. 16(a), where as the error tolerance value is decreased, BARTMAP is forced to create more F_2 nodes to compensate, resulting in steadily greater generalization performance. Note that the training error steadily decreases while the Rademacher penalty steadily increases along this same path (from right to left) in Fig. 16(a). Also, the low point in the sum of training error and Rademacher penality (the cross-over in Fig. 16(a)) is far away from the best generalization performance. This information also demonstrates that this particular learning problem is not a good candidate for structural risk minimization, as was shown in the previous section.

Fig. 16 (b) and (c) show that by adjusting the Boosted ARTMAP error tolerance value from $\varepsilon=0.0$ to $\varepsilon=0.5$, the performance of learning is significantly affected in both the training error achieved as well as the network complexity, as seen through the Rademacher penalty, on learning problems with noise and/or classification overlap. In these cases, the best network performance is achieved very near the point where the sum of training error and Rademacher penalty is minimized, however, because formal structural risk minimization is not used in BARTMAP, this characterization is not completely precise.

On the issue of selecting the error tolerance value for BARTMAP, Fig. 16(a)–(c) clearly show that a simple binary search on the values between 0 and 0.5 can quickly result in an error tolerance value with very good generalization performance. These particular problems have smooth generalization error surfaces, and so selection of the error tolerance parameter for BARTMAP is straightforward, but in cases where the error surface is not smooth, it might be difficult to find the optimal value. However, a simple check can be made by using a binary search across the values in {0, 0.1, 0.2, 0.3, 0.4, 0.5} or in {0, 0.05, 0.10.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5}. Also, because BARTMAP can reduce to Fuzzy ARTMAP, an error tolerance value of 0 will always show what Fuzzy ARTMAP can do. This can be seen at the far left in Fig. 16(a)–(c).

6.3. Rademacher penalty for noisy circle-in-the-square

In this section, the average Rademacher penalty is computed for each architecture on the noisy circle-in-the-square learning problem. Table 7 shows that most of the architectures have a Rademacher penalty near or greater than 0.5, which indicates a tendency for 'over-fitting' the training data. ART-EMAP was specifically designed to reduce the effect of noisy data, which it does very well, as indicated by the very low Rademacher penalty value and good performance on this learning problem in Table 2. However, this noise reduction is not available during training. ART-EMAP training is identical to Fuzzy ARTMAP, thus, even though it seems to have a low penalty value, its effective complexity penalty is the same as Fuzzy ARTMAP's.

PROBART also has a very good Rademacher penalty value with this learning problem, and its generalization performance is also seen to be very good in Table 2. The complexity of PROBART is controlled by tuning the vigilance value of its A-side Fuzzy ART sub-network. A larger value for this parameter can cause PROBART to create many F_2 nodes similar to the operation of Fuzzy ART with a high vigilance value. This value must be determined for each situation similar to the error tolerance in BARTMAP. Note that the Rademacher penalty value for μARTMAP and BARTMAP can be adjusted by adjusting their input parameters. In fact, BARTMAP will achieve a Rademacher penalty of zero by setting the error tolerance to 0.5, as shown in Fig. 16, but the training error, in general, will be too large using this error tolerance.

The Rademacher value in Table 7 for BARTMAP-SRM is the complexity penalty seen where its solution is achieved, that is at a complexity of 40 F_2 nodes. The complete Rademacher penalty for BARTMAP-SRM on this problem is the steadily increasing dotted line shown in Fig. 17(b).

6.4. Structural risk minimization and structural boosted ARTMAP

Fig. 17 shows the space of networks Structural Boosted ARTMAP (BARTMAP-SRM) used to produce its answer on four of the statistical learning problems used previously. Each point in Fig. 17(a)–(d) shows the performance and structural risk characteristics of a single network at a particular complexity value. The horizontal axis charts the number of F_2 nodes

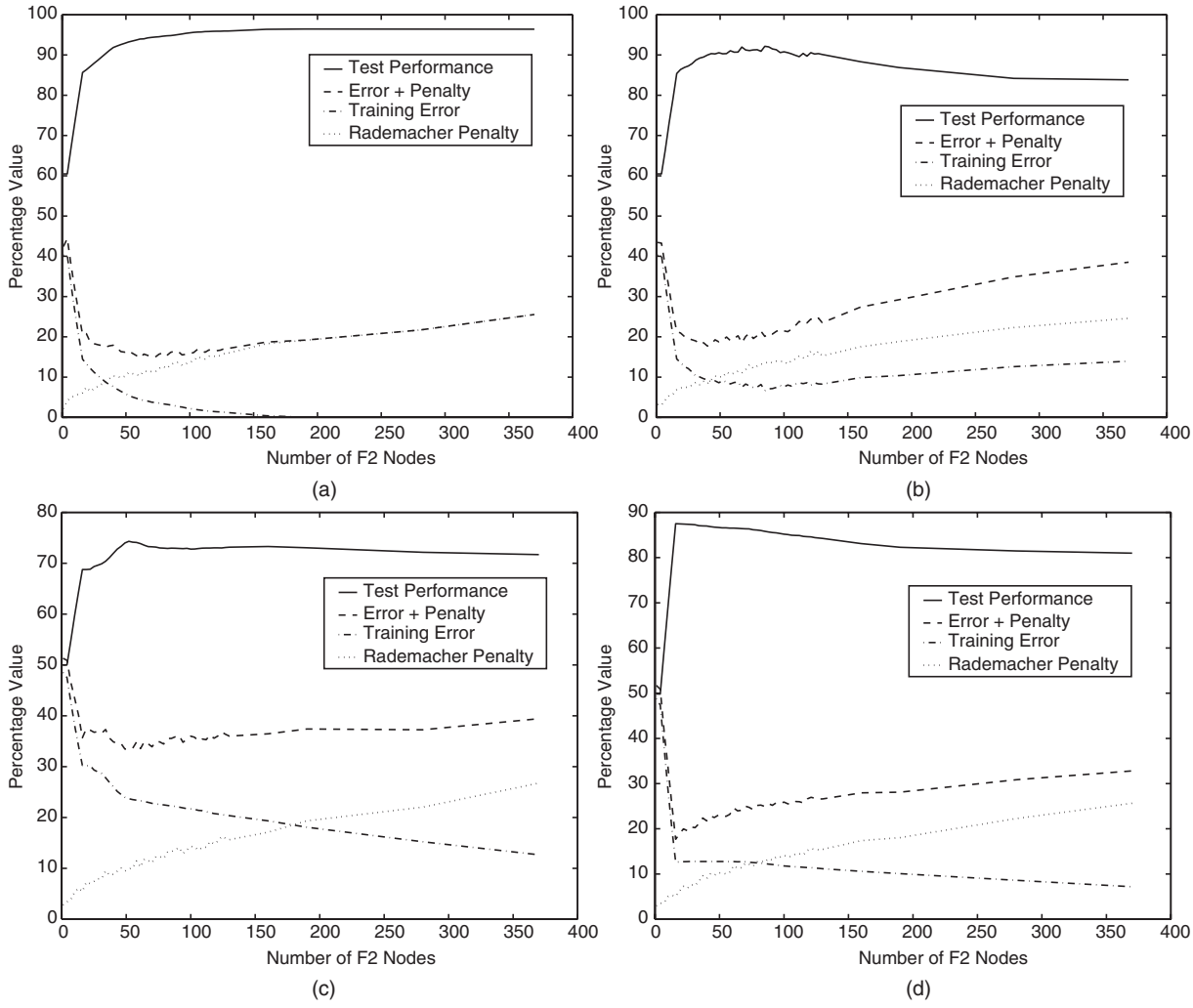

Fig. 17. The performance of Structural Boosted ARTMAP on the (a) circle-in-the-square, (b) noisy circle-in-the-square (c) overlapping circle-in-the-square and (d) overlapping squares problems.

used by a particular network. The vertical axis represents a percentage value for each network at a particular complexity. The solid line represents the test performance. The dash–dash line represents the computed value of training error plus Rademacher penalty. The dash–dot line represents the training error by itself, and the dot–dot line represents the Rademacher penalty by itself. Each of these networks is considered during BARTMAP-SRM learning, as can be seen in Fig. 12.

The actual solution output by BARTMAP-SRM is that network with the minimum combination of training error and Rademacher penalty. BARTMAP-SRM does not get to see the test performance, but rather that is shown here to compare the actual behavior of all of the networks that BARTMAP-SRM considers during learning. For this first experiment, BARTMAP-SRM's output network occurs at a complexity of 61 F_2 nodes, even though there are many networks with more F_2 nodes that have a higher test performance. This particular problem is an example where structural risk minimization is just not needed, although BARTMAP-SRM does output a good network. Empirical risk minimization *a la* Fuzzy ARTMAP does just fine. In addition, this result shows that BARTMAP-SRM will need an arbitrarily large number of these hyperboxes to approximate the circle precisely, as expected.

Fig. 17(b) is an example where structural risk minimization provides an advantage over empirical risk minimization. This plot shows that as the network complexity increases beyond a certain value, test performance decreases steadily. Over-fitting the data with too many F_2 nodes decreases generalization performance. Fig. 17(c) shows another example where structural risk minimization can be used to keep the network from becoming unnecessarily large. In Fig. 17(d), there is a steady and dramatic drop off in performance when the network complexity increases beyond 16. The reason for this is that BARTMAP-SRM only needs 16 hyperboxes to solve this problem exactly, and any extra F_2 nodes will only decrease its generalization performance. In each of these examples, the network output by BARTMAP-SRM, as determined by the crossover point of training error and Rademacher penalization,

has a test performance, which is very close to the optimal network test performance. In these situations, structural risk minimization benefits generalization performance.

7. Conclusions and future work

The experimental results in Section 6 demonstrate that Boosted ARTMAP offers an improvement over Fuzzy ARTMAP in learning situations where there is overlap between classes. Another benefit of Boosted ARTMAP is a reduction in the number of F_2 nodes needed during learning, at the expense of more epochs on the training data. This reduced hypothesis complexity results in improved generalization performance consistent with the theory of structural risk minimization for cases of classification overlap. In situations where there is no class overlap, Boosted ARTMAP reduces to Fuzzy ARTMAP with an error tolerance of zero. Anagnostopoulos and Georgiopoulos, 2002 has shown that the Boosted ARTMAP MAP field can be used with non-hyperbox extensions of Fuzzy ARTMAP, such as Ellipsoid ARTMAP. In fact, it is possible that the Boosted ARTMAP MAP field might be used in conjunction with Gaussian ARTMAP so that it can be allowed to reach a stable solution as opposed to stopping it after a set number of epochs. At present, the authors are continuing to research ways of bounding Boosted ARTMAP's generalization performance as well as Fuzzy ARTMAP's according to the data-driven analysis used by Koltchinskii (2001); Lazano, (2000).

The research presented in this paper was designed specifically to address learning in situations where the best possible generalization error is greater than zero. In these situations simply employing empirical risk minimization will result in a solution, which is not as good as can be obtained with structural risk minimization. The Rademacher penalty provides direct empirical evidence that Boosted ARTMAP's error tolerance parameter can be used to directly affect its structural complexity and thus indirectly affect its generalization performance.

Boosted ARTMAP stands on its own as a learning algorithm in noisy or overlapping data situations. Although Boosted ARTMAP does not perform formal boosting, it is a very reasonable first order approximation toward conducting boosting in a Fuzzy ARTMAP-based neural network. Fuzzy ARTMAP does not perform boosting because it always tries to learn each new sample precisely and it does not attempt to evaluate its learning in terms of generalization error or structural complexity. Boosted ARTMAP improves upon Fuzzy ARTMAP by not attempting to learn each sample precisely. Boosted ARTMAP also uses on-line learning, and thus it also does not evaluate its learning performance in terms of generalization error or structural complexity; however, the generalization performance and structural complexity can be adjusted using the Boosted ARTMAP error tolerance parameter. It remains an open problem as to whether or not formal boosting can be achieved using on-line learning with in a Fuzzy ARTMAP-based neural network.

Acknowledgements

Acknowledgments: This research was supported in part by a grant from the National Institutes of Health (NIBIB 1 R01 EB002618-01).

References

Anagnostopoulos, G. C., & Georgiopoulos, M. (2002). Ellipsoid ART and ARTMAP for incremental clustering and classification. In *Proceedings of the international joint conference on neural networks*.

Baras, J. S., & Dey, S. (1999). Combined compression and classification with learning vector quantization. *IEEE Transactions on Information Theory*, *45*(6), 1911–1920.

Blake, C. L., & Merz, C. (1998). *UCI repository of machine learning databases*. Department of Information and Computer Sciences, University of California, Irvine. http://www.ics.uci.edu/(mlearn/mlrepository.html.

Blumer, A., Ehrenfeucht, A., Haussler, D., & Warmuth, M. K. (1987). Occam's razor. *Information Processing Letters*, *24*, 377–380.

Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J. H., & Rosen, D. B. (1992). Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks*, *3*(5), 698–713.

Carpenter, G. A., Grossberg, S., & Rosen, D. B. (1991). Fuzzy ART: Fast stable learning and categorization of analog patterns by an adapative resonance system. *Neural Networks*, *4*(5), 759–771.

Carpenter, G. A., & Markuzon, N. (1998). ARTMAP-IC and medical diagnosis: Instance counting and inconsistent cases. *Neural Networks*, *11*, 323–336.

Carpenter, G. A., Milenova, B. L., & Noeske, B. W. (1998). Distributed ARTMAP: A neural network for fast distributed supervised learning. *Neural Networks*, *11*, 793–813.

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*(4), 805–818.

Dagher, I. (1997). *Properties of learning of the fuzzy ART neural network and improvements of the generalization performance of the fuzzy ARTMAP neural network*. PhD thesis, University of Central Florida, Orlando, FL.

Devroye, L., Györfi, L., & Lugosi, G. (1996). *A probabilistic theory of pattern recognition*. New York: Springer.

DiBenedetto, E. (2001). *Real analysis*. Boston: Birkhäuser.

Gómez-Sánchez, E., Dimitriadis, Y. A., Cano-Izquierdo, J. M., & López-Caronado, J. (2000). MicroARTMAP: Use of mutual information for category reduction in fuzzy ARTMAP. In *Proceedings of the international joint conference on neural networks, Como, Italy* (Vol. VI) (pp. 47–52).

Gómez-Sánchez, E., Dimitriadis, Y. A., Cano-Izquierdo, J. M., & López-Caronado, J., et al. (2002). MicroARTMAP: Use of mutual information for category reduction in fuzzy ARTMAP. *IEEE Transactions on Neural Networks*, *13*(1), 58–69.

Hush, D. (1997). *Learning from examples: From theory to practice. ICNN 97 Tutorial. Notes/viewgraphs*.

Kearns, M., & Mansour, Y. (1995). On the boosting ability of top-down decision tree learning algorithm. *AT&T Technical Report*.

Koltchinskii, V. (2001). Rademacher penalties and structural risk minimization. *IEEE Transactions on Information Theory*, *47*(5), 1902–1914.

Lazano, F. (2000). Model selection using rademacher penalties. In *Proceedings of second ICSC symposia on neural computation (NC2000)*. ICSC Academic Press.

Marriott, S., & Harrison, R. F. (1995). A modified fuzzy ARTMAP architecture for the approximation of noisy mappings. *Neural Networks*, *8*(4), 619–641.

Rudin, W. (1974). *Real and complex analysis* (2nd ed.). New York: McGraw-Hill.

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning*, *5*(2), 197–227.

van der Vaart, A. W., & Wellner, J. A. (1996). *Weak convergence and empirical processes*. New York: Springer.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. New York: Springer.

Vapnik, V. N. (1998). *Statistical learning theory*. New York: Wiley.

Vapnik, V. N., & Chervonenkis, A. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, *16*, 264–280.

Verzi, S. J. (2003). *Boosted ART and boosted ARTMAP: Extensions of fuzzy ART and fuzzy ARTMAP*. PhD thesis, University of New Mexico, Albuquerque, NM.

Verzi, S. J., Heileman, G. L., Georgiopoulos, M., & Anagnostopouleos, G. C. (2003). Universal function approximation with fuzzy ARTMAP. In *Proceedings of the international joint conference on neural networks*, *Portland, OR, USA*.

Vidyasagar, M. (1997). *A theory of learning and generalization*. New York: Springer.

Williamson, J. R. (1996). Gaussian ARTMAP: A neural network for fast incremental learning of noisy multidimensional maps. *Neural Networks*, *9*, 881–897.