Rademacher Penalization Applied to Fuzzy ARTMAP and Boosted ARTMAP

Stephen J. Verzi[†], Gregory L. Heileman[‡], Michael Georgiopoulus[*], Michael J. Healy[**]

[†]Computer Science Department, University of New Mexico, Albuquerque, NM 87131
verzi@cs.unm.edu

[‡]Department of Electrical & Computer Engineering, University of New Mexico, Albuquerque, NM 87131
heileman@eece.unm.edu

[*]Electrical & Computer Engineering, University of Central Florida, Orlando, FL 32816
mng@ece.engr.ucf.edu

[**]The Boeing Company, P.O. Box 3707 MS 7L-66, Seattle, WA 98124
michael.j.healy@boeing.com

Abstract

In our research we are interested in bounding the performance of Fuzzy ARTMAP and other ART-based neural network architectures, such as Boosted ARTMAP, according to the theory of Structural Risk Minimization. Structural risk minimization research indicates a trade-off between training error and hypothesis complexity. This trade-off directly motivated Boosted ARTMAP. In this paper, we present empirical evidence for Boosted ARTMAP as a viable learning technique, in general, in comparison to Fuzzy ARTMAP and other ART-based neural network architectures. We also show direct empirical evidence for decreased hypothesis complexity in conjunction with improved empirical performance for Boosted ARTMAP as compared with Fuzzy ARTMAP. Application of the Rademacher penalty to Boosted ARTMAP on a specific learning problem further indicates its utility as compared with Fuzzy ARTMAP.

I. Introduction

An important performance measure of a machine learning algorithm is its generalization capability. Generalization is characterized by the number of unseen examples correctly predicted by a learning algorithm given sample training data from which to learn. In this paper we focus on the particularly difficult situations in which the training data is drawn from pattern class distributions that are naturally overlapping or where noise in labeling is involved. For these types of problems, a learning algorithm must potentially deal with conflicting information in order to generalize to the underlying distributions.

Fuzzy ARTMAP is a neural network architecture for conducting supervised learning in a multidimensional setting [1], [2]. When Fuzzy ARTMAP is used on a learning problem, it is trained to the point that it correctly classifies all training data. This feature causes Fuzzy ARTMAP to "over-fit" some data sets, especially where the underlying pattern distributions have overlap. To avoid the problem of "over-fitting", we must allow for error during the training process. One solution for allowing error during training is to use a statistical approach. Such a method, proposed in this paper, is called Boosted ARTMAP.

In our research we are interested in bounding the performance of Fuzzy ARTMAP and other ART-based neural network architectures, such as Boosted ARTMAP, according to the theory of Structural Risk Minimization. Structural Risk Minimization research indicates a trade-off between training error and hypothesis complexity. This trade-off directly motivated our extension of Fuzzy ARTMAP into Boosted ARTMAP. In this paper, we present empirical evidence for Boosted ARTMAP as a viable learning technique, in general, in comparison to Fuzzy ARTMAP and other ART-based neural network architectures. We also show direct empirical evidence for decreased hypothesis complexity in conjunction with improved empirical performance for Boosted ARTMAP as compared with Fuzzy ARTMAP.

II. Structural Risk Minimization

The goal of learning is to find a hypothesis, $\hat{h}$, from a class of hypotheses, $\mathcal{H}$, with minimal generalization error

$$\hat{h} = \arg\min_{h \in \mathcal{H}} P\{h(x) \neq I_C(x)\}, \tag{1}$$

where C is the unknown target concept, $I_C(x)$ is the indicator function for C with arbitrary data sample x, and P is the probability mass function..

Structural risk minimization finds its roots in empirical risk minimization [3], [4], [5], [6], [7]. According to empirical risk minimization, a learner is given a set of labeled examples, $S = \{(x_1, y_1), ..., (x_m, y_m)\}$, where $x_i \in \mathcal{R}^d$ and $y_i \in \{0, 1\}$. The learner then finds a hypothesis, $\hat{h}$, from $\mathcal{H}$ with minimum empirical risk

$$\hat{h} = \arg\min_{h \in \mathcal{H}} \{L_m(h)\}$$

$$L_m(h) = \frac{1}{m} \sum_{j=1}^{m} I_{\{y_j \neq h(x_j)\}}(x_j). \tag{2}$$

The measure of empirical risk, $L_m(h)$ is also called training error.

In some cases, however, minimizing training error is not sufficient in finding a hypothesis with minimum generalization error. It is possible to find a hypothesis with minimum, even zero, training error that never-the-less has very poor generalization performance. Structural risk minimization was introduced by Vapnik [3], [7] to address the problems of empirical risk minimization by adding a penalty term

$$\hat{h}_{\hat{N}} = \arg \min_{h \in \mathcal{H}_{\mathcal{N}}, \mathcal{N} \geq \infty} \{L_m(h) + pen(m; N)\}. \quad (3)$$

The penalty term was included to bound the difference between the generalization error and training error by a function of the complexity of the hypothesis class, N,

$$P\{h(x) \neq I_C(x)\} \leq L(h) + |L(h) - L_m(h)|,$$
$$|L(h_N) - L_m(h_N)| \leq pen(m; N). \quad (4)$$

L is the training error and $pen(m; N)$ is a function of the complexity of the class of output hypotheses. Thus, there is a trade-off between training error and penalization where overall generalization error is greater than zero.

The penalty term can be bounded by the Vapnik-Chervonenkis (VC) dimension of the class of concepts [4]

$$pen(m; N) \leq K \sqrt{\frac{V(\mathcal{H}_{\mathcal{N}}) \log m}{m}}, \quad (5)$$

for some constant K. The VC dimension of a class of concepts is one measure of complexity for this set [8].

A. The Rademacher Penalty

Most penalty terms proposed for structural risk minimization rely heavily upon bounds that are abstracted away from the distribution of the problem data at hand. The Rademacher penalty was introduced by Koltchinskii [9] as a data-dependent penalty. 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.

Lozano [10] proposes a cleverly simple algorithm for computing the Rademacher penalty for a 0, 1-concept learner. In Lozano's method each sample (x_j, y_j) of a set of training data is randomly re-labeled with probability $\sigma_j = 0.5$ (i.e. with probability 0.5, y_j is flipped, either from 1 to 0 or visa versa), call it training set s_1. A second set of re-labeled data is immediately available by flipping all of the labels for s_1, call it training set s_2. Next, the learner is trained using both $s1$ and $s2$, separately, to produce two hypotheses, $h1$ and $h2$. The Rademacher penalty is then estimated as

$$pen(h_1) = \frac{1}{m} \sum_{j=1}^{m} \sigma_j I_{\{y_j \neq h_1(x_j)\}}(x_j), y_j \in s_1,$$
$$pen(h_2) = \frac{1}{m} \sum_{j=1}^{m} \sigma_j I_{\{y_j \neq h_2(x_j)\}}(x_j), y_j \in s_2,$$
$$pen(m, N) = \max(pen(h_1), pen(h_2)). \quad (6)$$

The Rademacher penalty, computed as above, allows us measure the complexity of a learner's hypothesis space, by determining how well it will satisfy, through learning, two very dis-similar training sets. Note that a learner which attempts to achieve 0 training error will produce a large Rademacher penalty, since it will attempt to satisfy two such dis-similar training sets exactly.

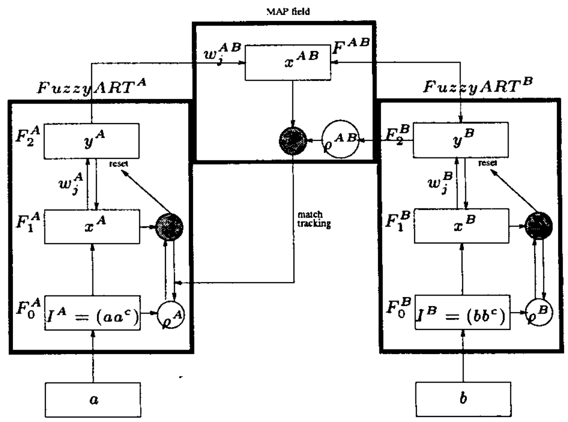

Fig. 1. The Fuzzy ARTMAP Architecture.

III. Fuzzy ARTMAP

Fuzzy ARTMAP is a neural network architecture designed to learn a mapping between example instances and their associated labels. Fuzzy ARTMAP is composed of two Fuzzy ART neural network modules connected through a MAP field [2]. During training, the pair (a, b) is presented to the neural network, where $a \in [0, 1]^d$ and $b \in \{0, 1, ..., C - 1\}$. In most cases, there will only be two classes, or labels, thus, $C = 2$ and $b \in \{0, 1\}$. The instance, a, is presented to the A-side Fuzzy ART module (ART^A) and b is presented to the B-side Fuzzy ART module (ART^B) in figure 1.

A. Fuzzy ART

The Fuzzy ART neural network architecture was designed to cluster real-valued data into categories [11]. Fuzzy ART is structured into three layers of interacting nodes, labeled F_0, F_1 and F_2, where the output of F_0 is connected to F_1, and F_1 and F_2 are mutually connected. At F_0, a d-length input vector from the environment is complement coded and passed on to F_1. The process of complement coding a pattern vector, a, produces a new vector $I^A = (a, a^c)$, where a^c is the complement of a. There are $2d$ nodes in layer F_1, and $N \geq 1$ nodes in layer F_2. The activation at node j of the F_2 layer, called $T_j(I)$, is computed as a weighted sum of I^A and the weights w_j, see Eq. 7 below. Note that these weights connect the F_1 layer to the F_2 of the Fuzzy ART module. The F_2 layer always has at least one node available which has not yet been trained, called the uncommitted node. The other $N - 1$ nodes in the F_2 layer have already been committed, having learned at least one training instance each. The F_2 layer is allowed to grow as necessary.

The F_1 and F_2 layers interact to choose an F_2 template that best matches the complement coded input vector according to:

$$J = \max_{0 \leq j \leq N} T_j(I), T_j(I) = \frac{|I \wedge w_j|}{\alpha + |w_j|}. \qquad (7)$$

The parameter α, called the choice parameter, is usually a small positive quantity, $\wedge$ is the element-wise vector *min* operator, and $|\cdot|$ is the L_1-norm of a vector. The choice J is confirmed if the vigilance criterion is not violated,

$$\frac{|I \wedge w_J|}{|I|} \geq \rho. \qquad (8)$$

The vigilance parameter, ρ in Eq. (8), is a user input between 0 and 1, where a value closer to 1 indicates desired tighter coupling within clustered patterns, and a value closer to 0 allows less coupling within clustered patterns, in a category. Note that at least one F_2 node, the uncommitted node, will always satisfy the vigilance criterion. The maximum F_2 template node satisfying the vigilance criterion is allowed to learn the input vector, a condition called *resonance*.

There are two stages in ART cluster formation. A winner-take-all strategy is employed in choosing the best matching cluster template in the F_2 layer according to Eq. (7). Next, a vigilance check is performed to ensure that learning the input pattern in the chosen cluster will not degrade the template below the vigilance as in Eq. (8). Initially all template weights are set to 1, and learning proceeds as follows

$$w_J^{(new)} = \beta(I \wedge w_J^{(old)}) + (1-\beta)w_J^{(old)}, \qquad (9)$$

where β is the learning parameter. In this paper we will set $\beta = 1$ which is a special case called *fast learning*. Note that learning only occurs at the winning F_2 node, J, during resonance.

An important feature of Fuzzy ART is that the F_2 layer is allowed to grow as needed for a particular problem. A pool of uncommitted template nodes is maintained. A single uncommitted template node is always allowed to compete with existing committed templates nodes according to Eq. (7).

B. The Fuzzy ARTMAP MAP field

The Fuzzy ARTMAP architecture in figure 1 consists of two Fuzzy ART modules connected through a MAP field. The ART^A module is given pattern data and the ART^B module is given label data for a given supervised learning task. The MAP field links pattern clusters with label clusters. Supervised learning is performed in Fuzzy ARTMAP by ensuring that each ART^A template is linked with only one ART^B template. For the classification tasks in this paper, the ART^B module will have one F_2 node for each label. Thus, a many-to-one association from patterns to labels is formed in the Fuzzy ARTMAP MAP field.

The Fuzzy ARTMAP architecture ensures the many-to-one mapping through the use of a match tracking lateral reset, see figure 1. During training for a specific pattern and label pair, (x, y), let J^A be the best choice F_2 node from the A-side ART module satisfying the vigilance criterion for ρ^A, and let K^B be the best choice F_2 node from the B-side ART module satisfying the vigilance criterion for ρ^B. If J^A is uncommitted, then no lateral reset can occur, and J^A will be associated with K^B in the MAP field during learning. If J^A is committed, then it is already associated with an ART^B F_2 node, call it K'. A lateral reset occurs when $K' \neq K^B$. During a lateral reset, the A-side vigilance parameter is temporarily increased to $\frac{T_{J^A}(\alpha+|w_{J^A}|)}{|I|} + \varepsilon$, where ε is some small constant greater than 0. After the network has resonated with x, the A-side vigilance is returned to its baseline value. The lateral reset is used in Fuzzy ARTMAP to ensure that each training pattern resonates with an A-side F_2 node associated with a B-side F_2 node that has learned the pattern's label. A complete presentation of all training patterns is called an *epoch*. After a bounded number of epochs, Fuzzy ARTMAP is guaranteed to reach 0 training error [12]. Note that during testing it is possible for a test pattern to choose the uncommitted node. In this case no B-side label prediction is possible.

The Fuzzy ARTMAP MAP field weights, w_{jk}, are used to control associations between A-side F_2 nodes and B-side F_2 nodes. For an uncommitted node, j, $w_{jk} = 1, \forall k$, meaning that j is not currently associated with any B-side node, and in fact it is available for future learning. For a committed node, j, $w_{jK} = 1$ and $w_{jk} = 0, \forall k \neq K$, where j has already been linked with B-side F_2 node K.

Notice that Fuzzy ARTMAP performs empirical risk minimization, but this is done at the expense of hypothesis complexity. In ART-based architectures, hypothesis complexity is measured by the number of F_2 nodes needed during training. The hidden layer nodes of Fuzzy ARTMAP (in the $F2$ layer) compute axis-parallel hyper- rectangles, but the process of training a Fuzzy ARTMAP network allows for 0 margin of training error. This fact implies that under certain situations Fuzzy ARTMAP can be made to require an arbitrarily large number of hidden layer ($F2$) nodes. Consider the case where we are interested in learning to distinguish between two overlapping, but continuous distributions. In the area of overlap are an arbitrarily large number of training examples, each of which can require its own hidden layer node for ARTMAP to train with 0 margin of training error. In this case the complexity of Fuzzy ARTMAP, i.e. the number of hidden layer nodes, grows as the number of training samples. Thus, Fuzzy ARTMAP will "over-fit" the training data, reducing its overall generalization error performance, in these cases.

We propose a modification to Fuzzy ARTMAP allowing for increased margin of training error decreasing the number of hidden layer nodes used for the purpose of increasing the overall generalization error performance, specifically on learning problems involving overlapping class distributions.

IV. Boosted ARTMAP

In our research, we are interested in increasing the generalization error performance of Fuzzy ARTMAP, and Fuzzy ART architectures, especially in situations where there is significant overlap between classes due to noise or other causes. The focus of our research in this paper involves a simplification of a modification to Fuzzy ARTMAP also called Boosted ARTMAP [13], [14]. In the current version of the Boosted ARTMAP neural network architecture, we connect two ART modules from figure 1 using the MAP field from the original Boosted ARTMAP [13], re-described here for clarity.

A. Modification to Fuzzy ARTMAP MAP field

In Boosted ARTMAP, we incorporate two changes to the Fuzzy ARTMAP MAP field. First, we allow F_2 nodes from the A-side ART module to associate with all F_2 nodes in the B-side. We also keep track of association frequencies between A-side and B-side nodes, similar to PROBART [15]. The MAP field weights are initially set to 0, $w_{jk} = 0, \forall j, k$. Consider a training sample, (x, y) presented to the network, assume that node J is chosen in the A-side, and node K is chosen in the B-side of the architecture. During learning in the MAP field, the associated weight value is increased by 1, $w_{JK} = w_{JK} + 1$. All other weight values remain the same. We then use these frequencies to gage the performance of each F_2 node in the A-side ART module, not done in PROBART. Our estimate for the performance error of a committed node, j is

$$e_j = 1 - \frac{\max\limits_{1 \leq k \leq C} w_{jk}}{\sum_{k=1}^{C} w_{jk}} \tag{10}$$

In order to bound the learning process, we use the frequency information gathered in the MAP field with the lateral reset match tracking mechanism described in the Fuzzy ARTMAP section above. The input error tolerance parameter, ϵ, is used to control the lateral reset. Again, 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} by 1 would increase J's estimated error performance, Eq. (10), to a value greater than ϵ, a lateral reset occurs. The lateral reset is precisely the same as described in the Fuzzy ARTMAP section above. In fact the operation of Boosted ARTMAP is exactly the same as Fuzzy ARTMAP described above except for the frequency estimation and lateral reset of the MAP field.

The Boosted ARTMAP neural network architecture has a couple of distinct advantages over the original Boosted ARTMAP [13]. First, the training error of a Boosted ARTMAP network is explicitly bounded by the input 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 at least as small as ϵ. An original Boosted ARTMAP network starts in a very erroneous state, near 50% error, and proceeds to reduce the training error towards ϵ. However, it may take many F_2 nodes and many training epochs for this network to achieve its goal. Learning for the current version of Boosted ARTMAP proceeds, similar to Fuzzy ARTMAP, in producing a trained network with at most ϵ training error. Finally, if ϵ is set to 0, then a Boosted ARTMAP network reduces exactly to a Fuzzy ARTMAP network. An advantage of Boosted ARTMAP is that it is trained on-line, and while finding the best ϵ value does require some tuning, it is highly related to the overlap in the data at hand. Thus, appropriate values for ϵ can be determined through off-line a priori data analysis.

V. Empirical Results

For our empirical results, we first compare the generalization performance of Boosted ARTMAP (BARTMAP) with Fuzzy ARTMAP (FuzARTMAP), ART-EMAP [16], ARTMAP-IC [17], Distributed ARTMAP (dARTMAP) [18], Gaussian ARTMAP (GARTMAP) [19], μARTMAP [20], [21], the original Boosted ARTMAP (called EmpARTMAP here) [13] and Hierarchical ARTMAP (HARTMAP) [22] on several learning tasks. Finally, we compute Rademacher penalties for all architectures for two of the learning problems.

A. Simple 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 the Fuzzy ART architecture could be used. In our experiments, each network was trained on 1000 training samples and tested with 10000 test samples. For each of the learning problems, we conducted 10 such training/testing scenarios for the average values reported in the tables below.

For all architectures, an ART^A baseline vigilance of 0.0 and ART^B baseline vigilance of 1.0 was used, and the MAP field vigilance was 1.0. In GARTMAP, we used γ values of 0.1 or 0.2, and we trained GARTMAP for 5 epochs for each learning problem. The EmpARTMAP network was trained using a step size of 0.1 for increasing the vigilance values. HARTMAP was trained using the same value for both the baseline training vigilance and the baseline testing vigilance, in both cases either 0.8 was used. BARTMAP was trained using the same parameter values as Fuzzy ARTMAP. For both EmpARTMAP and BARTMAP, the desired error tolerance values are problem specific.

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	4.4
ARTMAP-IC	7.0	24.7	95.9	0.6
dARTMAP	1.0	13.7	90.9	2.3
GARTMAP(0.1)	5.0	11.4	85.6	15.7
EmpARTMAP	9.3	125.5	85.2	3.5
HARTMAP(0.8)	2.4	126.4	89.7	1.2
μARTMAP(0.15)	49.6	17.0	93.3	3.2
BARTMAP(0.05)	8.3	19.6	94.9	0.5

TABLE I
CIRCLE-IN-THE-SQUARE.

Architecture	*Epochs*	F_2 *Nodes*	*% correct*	*std. dev.*
FuzARTMAP	7.5	202.6	73.0	1.9
ART-EMAP	7.5	202.6	79.4	6.8
ARTMAP-IC	7.5	202.6	72.9	1.4
dARTMAP	1.0	60.3	66.8	4.0
GARTMAP(0.2)	5.0	17.1	84.2	6.1
EmpARTMAP	9.5	147.9	82.5	3.9
HARTMAP(0.8)	2.4	126.4	86.4	1.9
μARTMAP(0.3)	47.3	30.0	69.1	9.0
BARTMAP(0.25)	13.3	63.8	85.3	2.0

TABLE II
NOISY CIRCLE-IN-THE-SQUARE.

Circle-in-the-Square [2]. In this problem, the circumference of the circle represents the optimal decision boundary. The area of the circular class is half that of the square, and both are centered about the same point. In table I, we see the learning performance of Fuzzy ARTMAP, GARTMAP ($\gamma = 0.1$), EmpARTMAP ($\epsilon = 0.1$), HARTMAP ($\rho = 0.8$) and BARTMAP ($\epsilon = 0.05$ on the circle-in-square problem averaged over the 100 experiments. The second column shows the average number of passes through the training data, i.e., epochs, needed to reach a solution. The third column give 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 10 experiments.

Noisy Circle-in-the-Square. In this problem, we use the training data from the previous problem, except that each label is flipped with probability of 0.2. In table II, we see the learning performance of Fuzzy ARTMAP, GARTMAP ($\gamma = 0.2$), EmpARTMAP ($\epsilon = 0.25$), HARTMAP ($\rho = 0.8$) and BARTMAP ($\epsilon = 0.25$ on the noisy circle-in-square problem averaged over the 10 experiments.

Overlapping Circle and Square. This experiment involves a uniformly distributed circle overlapping a uniformly distributed square, where the circle has half the area of the square, as in the circle-in-the-square problem above. Both circle and square are centered on the same point. At this point, we only have results for FuzARTMAP, GARTMAP, EmpARTMAP, HARTMAP, and BARTMAP, and these values were averaged over 100 such experiments.

Architecture	*Epochs*	F_2 *Nodes*	*% correct*	*std. dev.*
FuzARTMAP	7.7	176.3	68.8	0.8
GARTMAP(0.1)	5.0	12.8	67.4	9.4
EmpARTMAP	7.0	55.9	70.0	2.0
HARTMAP(0.8)	3.3	217.0	70.3	1.4
BARTMAP(0.35)	9.6	18.7	73.2	1.7

TABLE III
OVERLAPPING CIRCLE AND SQUARE.

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(0.2)	13.4	57.0
EmpARTMAP(0.25)	10.2	51.7
EmpARTMAP(0.38)	25.2	15.1
HARTMAP(0.8)	7.8	52.4
μARTMAP(0.3)	30.9	39.2
BARTMAP(0.25)	11.6	34.5

TABLE IV
RADEMACHER PENALTY FOR NOISY CIRCLE-IN-THE-SQUARE.

B. Rademacher Penalty for Noisy Circle-in-the-Square.

Here we compute the average Rademacher penalty each architecture produces for the Noisy circle-in-the-square problem. The results are averaged over 10 such experiments. In table IV we see 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 effect of noisy data, which it does very well, as indicated by the very low Rademacher penalty value. 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 Rademacher penalty is the same as Fuzzy ARTMAP's. Empirical ARTMAP has the capability of affecting the Rademacher penalty based upon its desired error tolerance parameter (ϵ, with values 0.38 and 0.25 shown in table IV).

A key feature of Boosted ARTMAP is its ability to

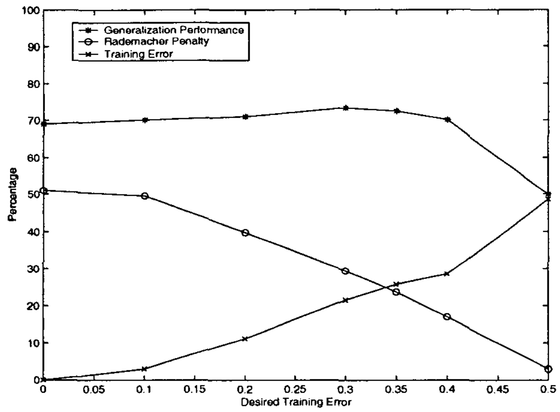

Fig. 2. Change in Performance for Boosted ARTMAP.

indirectly effect the Rademacher penalty through the value of its desired error tolerance parameter (ϵ). The use of ϵ in BARTMAP is much more pronounced that with EmpARTMAP as demonstrated in figure 2. In this figure, BARTMAP's performance is plotted versus increasing values for ϵ from 50 to 50000. In figure 2, we see exactly what structural risk minimization theory predicts, that the best generalization performance will occur where the the sum of both training error and penalization are minimized, occurring at $\epsilon = 0.35$, see table III.

VI. Conclusions and Future Work

After conducting the experiments, we have seen that Boosted ARTMAP is a reasonable alternative to Fuzzy ARTMAP in learning situations where there is overlap between classes. Another benefit that BARTMAP provides a mechanism by which the number of F_2 nodes necessary for learning is reduced, 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 and noisy data. Furthermore, In situations where there is no class overlap, Boosted ARTMAP can be made to execute exactly as Fuzzy ARTMAP by using a desired error tolerance of 0.

References

[1] Gail A. Carpenter, Stephen Grossberg, and David B. Rosen, "Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system," *Neural Networks*, vol. 4, no. 5, pp. 759–771, 1991.

[2] Gail A. Carpenter, Stephen Grossberg, Natalya Markuzon, John H. Reynolds, and David B. Rosen, "Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps," *IEEE Transactions on Neural Networks*, vol. 3, no. 5, pp. 698–713, 1992.

[3] Vladimir N. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, 1995.

[4] Luc Devroye, László Györfi, and Gábor Lugosi, *A Probabilistic Theory of Pattern Recognition*, Springer-Verlag, New York, 1996.

[5] Aad W. van der Vaart and Joh A. Wellner, *Weak Convergence and Empirical Processes*, Springer-Verlag, New York, 1996.

[6] Mathukumalli Vidyasagar, *A Theory of Learning and Generalization*, Springer-Verlag, New York, 1997.

[7] Vladimir N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, 1998.

[8] Vladimir N. Vapnik and A. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," *Theory of Probability and its Applications*, vol. 16, pp. 264–280, 1971.

[9] Vladimir Koltchinskii, "Rademacher penalties and structural risk minimization," *IEEE Transactions on Information Theory*, 2000, To appear.

[10] Fernando Lozano, "Model selection using rademacher penalties," in *Proceedings of Second ICSC Symposia on Neural Computation (NC2000)*. 2000, ICSC Academic Press.

[11] Gail A. Carpenter, Stephen Grossberg, and John H. Reynolds, "ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network," *Neural Networks*, vol. 4, no. 5, pp. 565–588, 1991.

[12] Michael Georgiopoulus, Juxin Huang, and Gregory L. Heileman, "Properties of learning in ARTMAP," *Neural Networks*, vol. 7, no. 3, pp. 495–506, 1994.

[13] Stephen J. Verzi, Gregory L. Heileman, Michael Georgiopoulus, and Michael J. Healy, "Boosting the performance of ARTMAP," in *Proceedings of IJCNN 98*, 1998, pp. 396–401.

[14] Stephen J. Verzi, Gregory L. Heileman, Michael Georgiopoulus, and Michael J. Healy, "Boosted ARTMAP," in *Proceedings of Systemics, Cybernetics and Informatics, SCI'2000*, 2000.

[15] Shaun Marriott and Robert F. Harrison, "A modified fuzzy ARTMAP architecture for the approximation of noisy mappings," *Neural Networks*, vol. 8, no. 4, pp. 619–641, 1995.

[16] Gail A. Carpenter and William D. Ross, "ART-EMAP: A neural network architecture for object recognition by evidence accumulation," *IEEE Transactions on Neural Networks*, vol. 6, no. 4, pp. 805–818, 1995.

[17] Gail A. Carpenter and Natalya Markuzon, "ARTMAP-IC and medical diagnosis: Instance counting and inconsistent cases," *Neural Networks*, vol. 11, pp. 323–336, 1998.

[18] Gail A. Carpenter, Boriana L. Milenova, and Benjamin W. Noeske, "Distributed ARTMAP: a neural network for fast distributed supervised learning," *Neural Networks*, vol. 11, pp. 793–813, 1998.

[19] James R. Williamson, "Gaussian ARTMAP: A neural network for fast incremental learning of noisy multidimensional maps," *Neural Networks*, vol. 9, pp. 881–897, 1996.

[20] Eduardo Gómez-S'anchez, Yannis A. Dimitriadis, Jos'e M. Cano-Izquierdo, and Juan L'opez-Coronado, "MicroARTMAP: use of mutual information for category reduction in Fuzzy ARTMAP," in *Proceedings of the International Joint Conference on Neural Networks, IJCNN2000*, Como, Italy, jul 2000, vol. VI, pp. 47–52.

[21] Eduardo Gómez-S'anchez, Yannis A. Dimitriadis, Jos'e M. Cano-Izquierdo, and Juan L'opez-Coronado, "μARTMAP: use of mutual information for category reduction in Fuzzy ARTMAP," *IEEE Transactions on Neural Networks*, 2001, To appear.

[22] Stephen J. Verzi, Gregory L. Heileman, Michael Georgiopoulus, and Michael J. Healy, "Hierarchical ARTMAP," in *Proceedings of the International Joint Conference on Neural Networks, IJCNN2000*, Como, Italy, jul 2000, pp. 41–46.