Off-line Structural Risk Minimization and BARTMAP-S

Stephen J. Verzi
Computer Science Department, University of New Mexico
Albuquerque, NM 87131, USA
verzi@eece.unm.edu
and
Gregory L. Heileman
Department of Electrical & Computer Engineering, University of New Mexico
Albuquerque, NM 87131, USA
heileman@eece.unm.edu
and
Michael Georgiopoulos
Electrical & Computer Engineering, University of Central Florida
Orlando, FL 32816, USA
mng@ece.engr.ucf.edu
and
Georgios Anagnostopoulos
Electrical & Computer Engineering, University of Central Florida
Orlando, FL 32816, USA
anagnostop@email.com

Abstract

BARTMAP-S introduced a neural network architecture with which structural risk minimization can be performed, although indirectly. BARTMAP-S as previously described is trained in an on-line fashion, consistent with the original way intended for the Fuzzy ARTMAP neural network architecture. Here we will propose an extension to BARTMAP-S for conducting off-line learning. Consequently, this alternate mode of learning will allow us to conduct structural risk minimization more directly. In this paper, we will describe the new architecture and present some empirical results to demonstrate the usefulness of structural risk minimization in learning with an ARTMAP-based neural network.

Keywords: Adaptive Resonance Theory, Machine Learning, Classification, Structural Risk Minimization, Empirical Risk Minimization, Overlapping Pattern Classes, Generalization Performance, Neural Networks.

1 Introduction

It has been previously shown that through the use of its desired error tolerance parameter ϵ, Simplified Boosted ARTMAP (BARTMAP-S) [1, 2] can be used to minimize both training error and network complexity [2]. In these results, ϵ was varied from its minimum value to its maximum value, and we could see the effect on network complexity through the Rademacher penalty value. This penalty value was calculated off-line, however, even though BARTMAP-S is trained on-line, similar to Fuzzy ARTMAP [3]. The next step is to design a new off-line version of BARTMAP-S that uses the Rademacher penalty during learning to bound its network complexity.

Before we describe the off-line version of BARTMAP-S (called BARTMAP-SRM), we will present a small overview of structural risk minimization. Then we will describe Fuzzy ARTMAP and BARTMAP-S briefly. Next, the new BARTMAP-SRM will be detailed as well as its foundation in probability theory. Finally we will present some simple empirical learning problems and compare results from BARTMAP-SRM with other contemporary ARTMAP-based architectures.

2 Risk Minimization

Structural risk minimization finds its roots in empirical risk minimization [4, 5, 6, 7, 8]. Thus, we will briefly describe empirical risk minimization and follow with a description of structural risk minimization.

2.1 Empirical Risk Minimization

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

$$h^* = \arg\min_{h \in \mathcal{H}} P\{h(x) \neq I_c(x)\}, \qquad (1)$$

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.

In empirical risk minimization, a learner is given a set of labeled examples, $S = \{(x_1, y_1), \cdots, (x_n, y_n)\}$, where $x_i \in \mathcal{R}^m$ and $y_i \in \{0, 1\}$. The learner then attempts to

0-7803-7278-6/02/$10.00 ©2002 IEEE

find a hypothesis, $h^* \in \mathcal{H}$, with minimum empirical risk

$$h^* = \arg\min_{h\in\mathcal{H}}\{L_n(h)\}$$
$$L_n(h) = \frac{1}{n}\sum_{j=1}^{n} I_{\{y_j\neq h(x_j)\}}(x_j). \quad (2)$$

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

2.2 Structural Risk Minimization

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 [4, 8] to address the problems of empirical risk minimization by adding a penalty term

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

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

$$P\{h(x)\neq I_C(x)\} \leq L(h) + |L(h) - L_n(h)|,$$
$$|L(h^N) - L_n(h^N)| \leq pen(n;N), \quad (4)$$

where $L(h)$ is the generalization error of h, h^N is a hypothesis of complexity no greater than N, and $pen(n;N)$ 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 0.

The penalty term can be bounded by the Vapnik-Chervonenkis (VC) dimension of the class of concepts (or hypotheses) [5]

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

for some constant K where $V(\mathcal{H}^N)$ is the VC dimension of $\mathcal{H}^N$. The VC dimension of a class of concepts is one measure of complexity for this set [9].

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 [10] 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 proposes a cleverly simple algorithm for computing the Rademacher penalty for a "0 − 1"-concept learner [11]. In this method, each training sample (x_j, y_j) is randomly re-labeled with probability 0.5 (i.e. with probability 0.5, $\sigma_j = -1$ and y_j is flipped either from 1 to 0 or visa versa, otherwise $\sigma_j = 1$ and y_j is left alone)[1], call it training set s_1. Note that the σ_j's are Rademacher random variables. A second set of re-labeled data is immediately available by flipping all of the labels of s_1, call it training set 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

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

The Rademacher penalty, as computed in Eq. (6), 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.

3 Fuzzy ARTMAP

Fuzzy ARTMAP is a neural network architecture designed to learn a mapping between example instances and their associated labels. These training examples are denoted (x, y), where $x \in [0,1]^m$ is an example instance, and $y \in \{0, 1, ..., C-1\}$ is its corresponding label. In most cases, there will only be two classes, or labels, thus, $C = 2$ and $y \in \{0, 1\}$. Fuzzy ARTMAP [3] is composed of two Fuzzy ART neural network modules connected through a MAP field, as shown in Fig. 1. The instance, x, is presented to the A-side Fuzzy ART module (ART^A) and y is presented to the B-side Fuzzy ART module (ART^B). 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 ART^A (x) and ART^B (y) (see Fig. 1 below), maps to a single cluster template in the respective Fuzzy ART module. Then each ART^A cluster template is mapped to a single ART^B cluster template through the Fuzzy ARTMAP MAP field. The overall mapping learned by Fuzzy ARTMAP is a composition of these two separate mappings. During training, the pair (x, y) is preprocessed, with complement coding, to form the pair $((x, x^c), (y, y^c))$, where x^c is the complement of x, which

[1] Rademacher penalization and structural risk minimization apply to learning situations where there are more than two classes, but in this paper we will be dealing with two class learning.

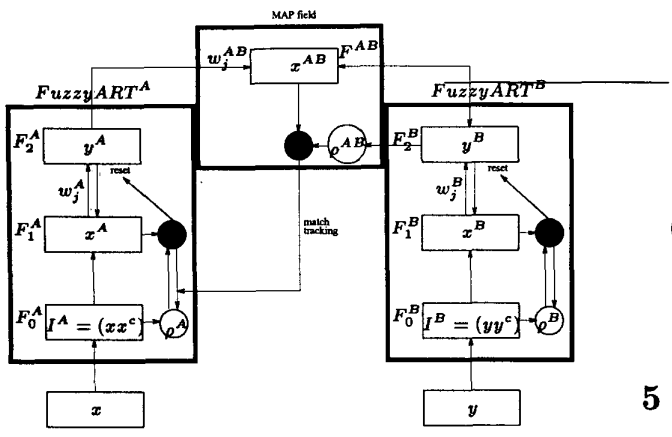

Figure 1: The Fuzzy ARTMAP Architecture.

is then presented to the neural network. Fuzzy ARTMAP performs supervised learning on (x, y), a data pattern and a label for that data pattern respectively.

4 BARTMAP-S

BARTMAP-S was designed to address performance difficulties of Fuzzy ARTMAP, especially in situations where there is significant overlap between classes due to noise or other causes. The BARTMAP-S network involves a simple modification to the Fuzzy ARTMAP MAP field.

Modified MAP Field. The BARTMAP-S architecture incorporates two changes to the Fuzzy ARTMAP MAP field. First each F_2 node from the A-side ART module is allowed to simultaneously associate with all F_2 nodes in the B-side. The association frequencies between A-side and B-side nodes are stored in the MAP field similar to PROBART [12]. BARTMAP-S bounds the learning process by using the frequency information gathered in the MAP field, in place of the vigilance test, with the lateral reset match tracking mechanism of Fuzzy ARTMAP. The input error tolerance parameter, ϵ, is used to control the lateral reset. The error tolerance takes on values between 0 and 0.5.

The application of the lateral reset mechanism, in BARTMAP-S, is precisely the same as in Fuzzy ARTMAP. In fact the performance of BARTMAP-S is exactly the same as Fuzzy ARTMAP, except for the use of frequency estimation during lateral reset of the MAP field and the accumulation of such frequency information during learning. Moreover, BARTMAP-S reduces precisely to Fuzzy ARTMAP when $\epsilon = 0$.

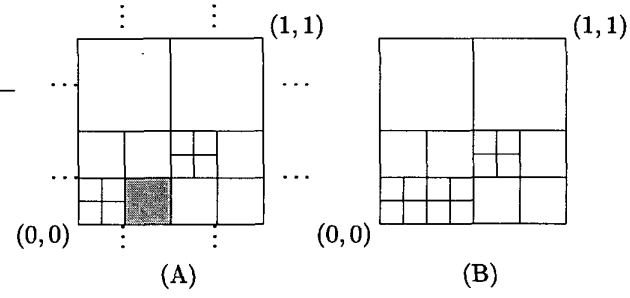

Figure 2: Some boxes belonging to Ω.

5 Axis Parallel Hyper-rectangles and Open Sets in $\mathcal{R}^m$

An interesting property of open sets in $\mathcal{R}^m$ is that each such non-empty open set is composed of a countable union of disjoint boxes belonging to $\Omega = \Omega_1 \cup \Omega_2 \cup \Omega_3 \cup \cdots$ [13]. Here, Ω_n is defined as the collection of all 2^{-n} boxes with corners at P_n, where P_n is the set of all $x \in \mathcal{R}^m$ whose coordinates are integer multiples of 2^{-n}. This property means that open sets in $\mathcal{R}^m$ can be covered by a collection of sets from Ω. It may take an infinite number of these boxes to cover a specific open set, but these will be countably infinite. In Fig. 2A, we see a collection of boxes from Ω_1, Ω_2, Ω_3, and Ω_4 for $\mathcal{R}^2$.

6 BARTMAP-SRM

In this paper, we are proposing a new off-line Fuzzy ARTMAP-based neural network architecture which employs structural risk minimization with Rademacher penalization. Our new architecture, called BARTMAP-SRM is a very simple modification of BARTMAP-S. In BARTMAP-SRM, we will employ off-line learning using a series of BARTMAP-S networks which are not allowed to grow (i.e., learning is turned off). This algorithm was motivated by the hyper-boxes in Ω previously described.

BARTMAP-SRM begins with a BARTMAP-S network complexity of 4 nodes[2]. There is one F_2 node for each half of each dimension in cross product. This network is then tested upon the entire training set. The BARTMAP-S MAP field has the training error for each F_2 node stored in it. The F_2 node with the greatest error contribution is then split into 4 new F_2 nodes and it is replaced by the same nodes as they cover the same space as the original hyper-box. The grayed box in Fig. 2A is split into 4 new boxes in Fig. 2B.

As far as algorithm termination, BARTMAP-SRM need not create more F_2 nodes than there are training

[2]In general this value will be m^2 for m dimensions. In this paper, however, m is always 2. Throughout the rest of the description we will let $m = 2$ for simplicity, but our algorithm does scale up to m dimensions.

samples. Therefore, after each network is tested upon the training data, BARTMAP-SRM can compute the Rademacher penalty, using the same training data, with the existing complexity level (i.e. N = the number of F_2 nodes in Eq. 6). In most situations BARTMAP-SRM need not consider networks with complexity higher than one-half or one-third of the number of training samples. To output its solution, BARTMAP-SRM chooses the network with minimum combined training error plus Rademacher penalty, of all those networks considered.

7 Empirical Results

For our empirical results, we compare the generalization performance of BARTMAP-SRM with Fuzzy ARTMAP, BARTMAP-S, ART-EMAP [14], ARTMAP-IC [15], Distributed ARTMAP (dARTMAP) [16], Gaussian ARTMAP (GARTMAP) [17], Boosted ARTMAP (BARTMAP) [18], Micro ARTMAP (μARTMAP) [19, 20], Hierarchical ARTMAP (BARTMAP-H) [21], and further modification to BARTMAP-S called Classification Boosted ARTMAP (BARTMAP-C).

In each of the following learning problems, one class was labeled 0 and the other 1. All data were normalized to fit within the unit square for Fuzzy ART complement coding. Also, each class contributed equally to both the training and test data sets. Each network was trained using 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.

An ART^A baseline vigilance of 0.0 and ART^B baseline vigilance of 1.0 were used throughout. A MAP field vigilance of 1.0 was used for those architectures that use this parameter. In all experiments, BARTMAP-H was trained with $\rho = 0.8$. In each of the tables below, the second column shows the average number of passes through the training data, i.e., 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 performance percentage across the 10 experiments.

7.1 Circle-in-the-Square [3].

In this problem, the circumference of the circle represents the optimal decision boundary. The diameter of the circular class is equal in size to the hypotenuse of a square half the size of the big square, and both are centered about the same point. In table 1, Fuzzy ARTMAP holds the benchmark. ART-EMAP looses some performance due its suppression of small isolated data pockets. BARTMAP-S reduces exactly to Fuzzy ARTMAP with $\epsilon = 0$.

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$_{\lambda=.1}$	5.0	11.4	85.6	16.5
BARTMAP$_{\epsilon=.1}$	9.3	125.5	85.2	3.7
μARTMAP$_{h=.15}$	49.6	17.0	93.3	3.4
BARTMAP-H	2.4	126.4	89.7	1.3
BARTMAP-S$_{\epsilon=.0}$	7.0	24.7	95.9	0.6
BARTMAP-C$_{\epsilon=.0}$	4.5	20.9	95.3	0.4
BARTMAP-SRM	20.0	61.0	94.0	0.6

Table 1: Circle-in-the-Square.

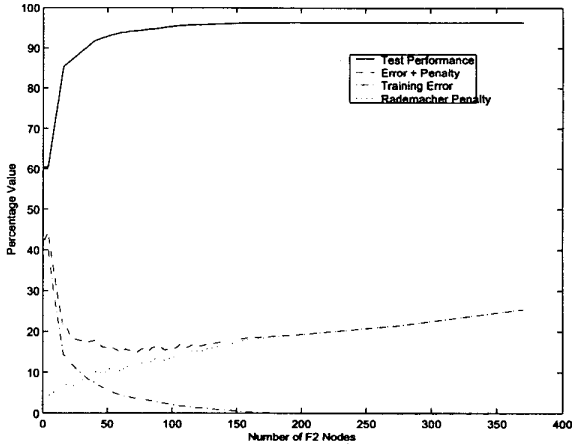

Figure 3: BARTMAP-SRM and Circle-in-the-Square.

In Fig. 3, we see the space of networks BARTMAP-SRM used to produce its answer. The bottom axis charts the number of F_2 nodes used by a particular BARTMAP-SRM network. This axis is the complexity of the network. The vertical axis represents a percentage value for each network at a particular complexity. The first line (solid) represents the test performance. The second line (dash-dash) represents the computed value of training error plus Rademacher penalty. The third line (dash-dot) represents the training error by itself, and the last line (dot-dot) represents the Rademacher penalty by itself. Figs. 3, 4, and 5 show all of the networks considered by BARTMAP-SRM internally during its learning, but 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 so that we can see the actual behavior of all of the net-

Architecture	*Epochs*	F_2 *Nodes*	% *correct*	*std. dev.*
FuzARTMAP	7.5	202.6	73.0	2.0
ART-EMAP	7.3	199.1	78.4	7.1
ARTMAP-IC	7.4	203.3	72.9	1.4
dARTMAP	1.0	57.8	68.0	4.8
GARTMAP$_{\lambda=.2}$	5.0	17.1	84.2	6.4
BARTMAP$_{\epsilon=.25}$	9.5	147.9	82.4	4.1
μARTMAP$_{h=.25}$	112.9	30.8	65.6	7.8
BARTMAP-H	2.4	126.4	86.4	2.0
BARTMAP-S$_{\epsilon=.25}$	13.3	63.8	85.3	2.1
BARTMAP-C$_{\epsilon=.25}$	6.4	46.1	84.0	3.4
BARTMAP-SRM	13.0	40.0	90.0	1.1

Table 2: Noisy Circle-in-the-Square.

works 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 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.

7.2 Noisy Circle-in-the-Square.

In this problem, we add 20% label noise to the previous learning problem. Thus with probability $\frac{1}{5}$ each sample label is flipped. This label noise is significant, but it does allow us to see the performance of the learning algorithms in the presence of noise. In table 2, μARTMAP does not handle noisy data very well. GARTMAP shows good performance throughout our experiments, 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. BARTMAP-H deals with noisy data by creating a hierarchy of cluster with greater specificity (usually less error) as it goes down its tree. BARTMAP-H tends to use many more F_2 nodes, in general, than the other algorithms.

In Fig. 4, we see our first example where structural risk minimization buys us something. This plot is a good example where as the network complexity increases, after a certain point, test performance decreases steadily. Over-fitting the data with too many F_2 nodes decreases generalization performance.

7.3 Overlapping Squares.

This experiment involves a uniformly distributed square overlapping a uniformly distributed square, where the

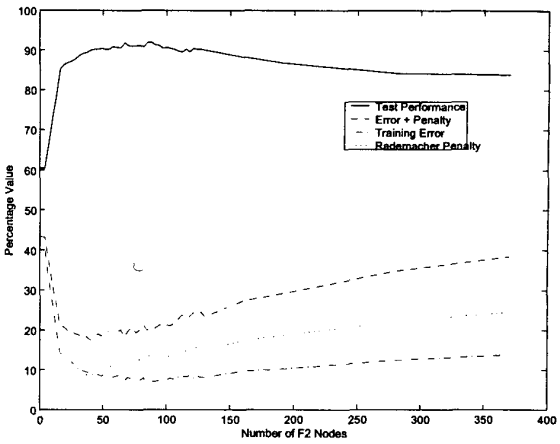

Figure 4: BARTMAP-SRM and Noisy Circle-in-the-Square.

Architecture	*Epochs*	F_2 *Nodes*	% *correct*	*std. dev.*
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$_{\lambda=.1}$	5.0	10.8	81.9	1.7
BARTMAP$_{\epsilon=.2}$	9.0	99.9	78.4	4.1
μARTMAP$_{h=.4}$	24.4	52.7	81.2	1.9
BARTMAP-H	2.6	114.0	78.3	1.0
BARTMAP-S$_{\epsilon=.25}$	9.3	20.8	83.3	2.1
BARTMAP-C$_{\epsilon=.5}$	2.0	2.0	87.0	0.4
BARTMAP-SRM	5.0	16.0	87.5	0.3

Table 3: Overlapping Squares.

smaller square has half the area of the larger square. Both squares are centered on the same point. In table 3, we see a problem that should be easy for the methods, using hyper-box F_2 nodes, to capture in very few nodes. Note that BARTMAP-C uses only 2 hyper-boxes here, and BARTMAP-SRM achieves a nearly optimal solution with exactly 16 hyper-boxes.

In Fig. 5, there is a steady drop off in performance when the network complexity increases beyond 16. The reason for this is that BARTMAP-SRM only needs 16 hyper-boxes to solve this problem exactly, and any more decrease its generalization performance.

8 Conclusions

In this paper we have shown through several simple learning problems how structural risk minimization can be

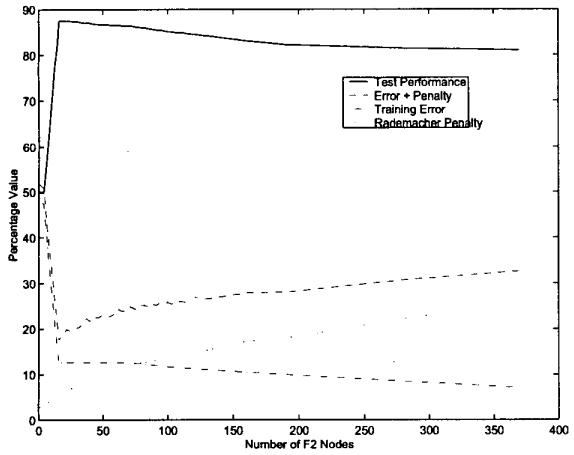

Figure 5: BARTMAP-SRM and Overlapping Squares.

helpful in providing solutions with greater generalization performance. The spirit of Occam's Razor [22] infers that we should not complicate things unnecessarily, and the Rademacher penalty is one measure of complexity useful in allowing us to achieve greater simplicity. Our future research will continue to push the use of structural risk minimization in ARTMAP-based learning, even on-line learning, if possible.

References

[1] Stephen J. Verzi, Gregory L. Heileman, Michael Georgiopoulos, and Michael J. Healy, "Boosting in ARTMAP networks," in *Proceedings of Systemics, Cybernetics and Informatics, SCI'2000*, 2000, pp. 473–478.

[2] Stephen J. Verzi, Gregory L. Heileman, Michael Georgiopoulos, and Michael J. Healy, "Rademacher penalization applied to Fuzzy ARTMAP and Boosted ARTMAP," in *Proceedings of the International Joint Conference on Neural Networks, IJCNN2001*, Washington DC, USA, july 2001, pp. 1191–1196.

[3] 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.

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

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

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

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

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

[9] 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.

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

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

[12] 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.

[13] Walter Rudin, *Real and Complex Analysis*, McGraw-Hill, New York, second edition, 1974.

[14] 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.

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

[16] 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.

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

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

[19] Eduardo Gómez-Sánchez, Yannis A. Dimitriadis, José M. Cano-Izquierdo, and Juan López-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.

[20] Eduardo Gómez-Sánchez, Yannis A. Dimitriadis, José M. Cano-Izquierdo, and Juan López-Coronado, "μARTMAP: use of mutual information for category reduction in Fuzzy ARTMAP," *IEEE Transactions on Neural Networks*, vol. 13, no. 1, pp. 58–69, jan 2002.

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

[22] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth, "Occam's razor," *Information Processing Letters*, vol. 24, pp. 377–380, 1987.