# Hierarchical ARTMAP

Stephen J. Verzi[†], Gregory L. Heileman[‡], Michael Georgiopoulos[*], 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
mjhealy@boeing.com

**Abstract**

We present a modification to the Fuzzy ARTMAP neural network architecture for conducting classification in a probabilistic setting. We call this new architecture Hierarchical ARTMAP (HARTMAP). Performance comparisons with Fuzzy ARTMAP, Gaussian ARTMAP and Boosted ARTMAP on some simple two-class problems are discussed. Experimental results indicate that HARTMAP yields better generalization results on problems involving overlap of the underlying pattern distributions.

## 1 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 situation in which the training data is drawn from pattern class distributions that are naturally overlapping. 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 those in which the underlying pattern distributions have overlap. To avoid the problem of "over-fitting", we must allow for error in the training process. One solution for allowing error during the training is to use a statistical approach. Such an approach is used in Gaussian ARTMAP [3] and in Boosted ARTMAP [4].

In this paper, we will present an extension to Boosted ARTMAP which uses a hierarchical structure of classification templates. Our architecture, called HARTMAP, was motivated by the desire to improve generalization performance using statistical methods, while maintaining a structure within which we can operate so that "difficult" portions of the problem space can be handled with adequate knowledge of consequences. The hierarchical structure allows us, at each level of the hierarchy, to estimate class frequencies so that we can determine parts of the problem space that are going to be difficult to properly classify. We are then free to deal with these hard areas by breaking them up into smaller hierarchies.

A summary of the paper is as follows. In section 2, we briefly describe some ART-based neural architectures: Fuzzy ARTMAP, Gaussian ARTMAP and Boosted ARTMAP. In section 3, we describe our extension, hierarchical ARTMAP. Empirical results are presented in section 4, and conclusions are discussed in section 5.

## 2 ART-based Architectures

The Fuzzy ART neural network architecture was designed to cluster data into categories [5]. Fuzzy ART is structured into three layers of interacting neural 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 $F_0$, an $M$-length input vector from the environment is complement coded and passed on to $F_1$.



0-7695-0619-4/00 $10.00 © 2000 IEEE

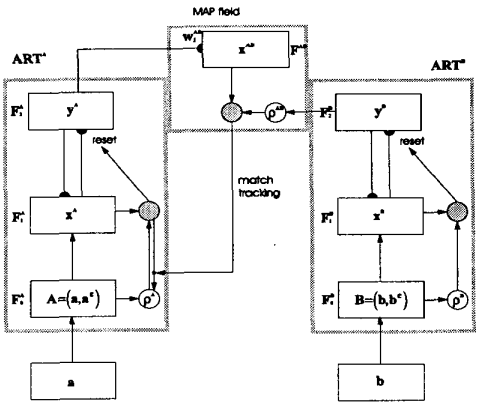


Figure 1: The Fuzzy ARTMAP Architecture.

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

$$T_j(I) = \frac{|I \wedge w_j|}{\alpha + |w_j|}. \tag{1}$$

This choice is confirmed if the vigilance criterion is not violated, i.e.,

$$\frac{|I \wedge w_j|}{|I|} \geq \rho. \tag{2}$$

The parameter $\alpha$, called the choice parameter, is usually a small positive quantity. The vigilance parameter, $\rho$, 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.

The process of complement coding a pattern vector, $a$, produces a new vector $A = (a, a^c)$, where $a^c$ is the complement of $a$. 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 given a complement coded input vector according to (1). 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 (2). Initially all template weights are set to 1, and learning proceeds as follows

$$w_j^{(new)} = \beta \left( I \wedge w_j^{(old)} \right) + (1 - \beta) w_j^{(old)},$$

where $\beta$ is the learning parameter. In this paper we will set $\beta = 1$ which is a special case called fast learning.

An important feature of ART is that the $F_2$ layer is allowed to grow as needed for a particular problem. A pool of templates is maintained, where a committed template has, at some point, learned at least one input pattern. One uncommitted template is allowed to compete with existing committed templates according to (1), given the initial weight settings.

**Fuzzy ARTMAP [2].** The 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. Thus, a many-to-one association from patterns to labels can be formed.

**Gaussian ARTMAP [3].** This architecture is a modification of Fuzzy ARTMAP proposed by Williamson. The structure of Gaussian ARTMAP is very similar to Fuzzy ARTMAP (figure 1), except that the $ART^A$module is replaced by a $GaussianART^A$ module, and no complement coding is done here. The Gaussian ARTMAP choice function is computed as follows:

$$T_j(I) = \frac{n_j}{\prod_{i=1}^{M}(\sigma_{ij}) \sum_{k=1}^{N}(n_k)} e^{-0.5 \sum_{i=1}^{M} (\frac{I_i - \mu_{ij}}{\sigma_{ij}})^2}. \tag{3}$$

In equation 3, $n_j$ is the number of training instances that have resonated with $F_2$ node $j$, $N$ is the number of currently committed $F_2$ nodes, and $\mu_{ij}$ and $\sigma_{ij}$ are the empirically estimated mean and variance for the $j$th $F_2$ node in dimension $i$. The Gaussian ARTMAP vigilance criterion function is:

$$e^{-0.5\sum_{i=1}^{M}(\frac{I_i-\mu_{ij}}{\sigma_{ij}})^2} \geq \rho. \tag{4}$$

Finally, the Gaussian ARTMAP update formulas for $F_2$ node $j$ in dimension $i$ are:

$$\begin{aligned} n_j &= n_j + 1 \\ \mu_{ij} &= (1 - \frac{1}{n_j})\mu_{ij} + \frac{1}{n_j} I_i \\ \sigma_{ij} &= \sqrt{(1 - \frac{1}{n_j})\sigma_{ij}^2 + \frac{1}{n_j}(I_i - \mu_{ij})^2} \end{aligned} \tag{5}$$

An extra parameter to Gaussian ARTMAP, $\gamma$ is used to set initial as well as minimum variance values (i.e. $\sigma_{ij} \leq \gamma$).

**Boosted ARTMAP (BARTMAP) [4].** This architecture is also structured similar to Fuzzy ARTMAP (figure 1), except that each $F_2$ node is given its own vigilance parameter. BARTMAP uses an extension of the MAP field from PROBART [6] to keep track of class frequency estimates for each $F_2$ node.

# 3 Hierarchical ARTMAP

While working with the BARTMAP architecture, the authors were struck by the need to determine the consequences of allowing more than one association between $BoostedART^A$ $F_2$ nodes and $BoostedART^B$ $F_2$ nodes. In Fuzzy ARTMAP, each data template ($A$-side) is allowed to associate with only one label template ($B$-side). BARTMAP data templates can associate with any number of label templates, and thus, the predicted label for a data template becomes the label template with the highest association. In this model, however, it is difficult to determine the consequences of using the maximally associated label template for each data template. In designing HARTMAP, we wanted to maintain an entire hierarchy of data template nodes, where at each level, we can determine how much estimated error we are willing to allow, and whether or not we are willing to use more data template nodes to reduce the estimated error.

Just as with the other ART-based neural architectures, Hierarchical ARTMAP is composed of two ART-like modules connected by a MAP field. The *HierarchicalART* (*HART*) module is an extension of the *BoostedART* (*BART*) module for conducting hierarchical classification of input data. What we mean by hierarchical classification is that levels of categorization will be formed as the network is trained. In our network architecture, each successive level will classify more tightly coupled groups of data. The hierarchy of categories allows us to train our network until we have achieved a specified level of accuracy, in training, or until all inputs are correctly classified. One benefit of HARTMAP is that it works on-line.

The *HART* extends the *BART* module by adding a hierarchical structure to the $F_2$ layer of nodes. By this we mean that each $F_2$ node is the "parent" of 0 or more "child" $F_2$ nodes. HARTMAP uses the same choice competition and vigilance criterion as Fuzzy ARTMAP as well as the same weight update functions (also used in Boosted ARTMAP [4]). The difference is that only a selected number of nodes is allowed to compete in each level of the hierarchy. Also in HARTMAP, we introduce two new input parameters to the system. The training baseline vigilance controls the depth to which the architecture is trained, and the testing baseline vigilance controls the depth to which a predicted label is sought.

The rules of operation are very straight forward. Each parent's children compete for resonance with an input instance. If the winning "child" node satisfies its vigilance criterion, then it becomes the new parent and we continue down the hierarchy. If this child node fails its vigilance test, then it is reset and the competition continues amongst its siblings. If none of the children satisfy their vigilance criteria, then a new node is created with a vigilance value greater than its parent, and it becomes the new parent. During training, descent into the hierarchy continues until an $F_2$ node is reached that wins its sibling competition, satisfies its vigilance criterion and its own vigilance is greater than training baseline vigilance. During testing,

descent continues in the same fashion until an $F_2$ node is found where its own vigilance is greater than or equal to the testing baseline vigilance.

At each level in the hierarchy, all children have a higher vigilance than their parent. This means that each of the children can only resonate with a portion of the input instances that their parent can resonate with. Note that we also have a top level node which does not compete in any choice competition, but rather manages its children. This top level node has a vigilance of 0, which means that all input instance could resonate with it. In training HARTMAP, we can learn the training data to 0% error by setting the training baseline vigilance to 1, but it will require at least as many $F_2$ nodes as training instances.

# 4 Empirical Results

For our empirical results, we compare the generalization performance of HARTMAP with Fuzzy ARTMAP as well as Gaussian ARTMAP and BARTMAP. The first learning problem consists of two overlapping 2D Gaussian distributions with the same means and different variances, and the second learning problem consists of two overlapping 2D Gaussian distributions with different means and the same variances. The third learning problem consists of two overlapping 2D Uniform distributions with the same centers and different boundaries (similar to problem one). Next, we trained each of the four networks on a learning problem consisting of two bimodal 2D Gaussian distributions [7]. Finally, we trained each of the four networks on the BUPA liver disorder problem from the UCI repository [8].

In each of the learning problems, one class was labeled 0 and the other 1, to allow for concept learning. All data were normalized to fit within the unit square so that the Fuzzy ART architecture could be used. Also, each class contributed equally to both the training and test data sets.

For the 2D generated data in our experiments, each network was trained on 1000 training samples and tested with either 1000 (bimodal 2D Gaussian learning problem) or 10000 (other 2D learning problems) test samples. For the UCI learning problems, each of the databases was sampled into 2/3 training/1/3 test sets. For each of the learning problems, we conducted 100 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 was used for Fuzzy ARTMAP, and the MAP field vigilance was 1.0. In Gaussian ARTMAP, we used $\gamma$ values of 0.01 or 0.1, and we ran Gaussian ARTMAP for 5 epochs for each learning problem. BARTMAP was run using 0.1 as a starting value for $BART^A$ vigilance values, and 0.1 was also used as a step size for increasing these values. A vigilance of 1.0 was used in BARTMAP for $ART^B$. BARTMAP was executed to an error tolerance of 0.31 for distribution 1, 0.1 for distribution 2, 0.2 for the uniform learning problem, 0.25 for the bimodal 2D Gaussian learning problem and 0.4 for the BUPA dataset.

HARTMAP was trained with training and testing baseline vigilance values of 0.8 except for the BUPA dataset where 0.1 was used. We used a vigilance step of 0.1 from parent to child in all of the learning problems in this paper. This means that each child node has a vigilance value of 0.1 greater than its parent.

**Overlapping Gaussians–Case 1.** Our first experiment is a difficult problem where one 2D Gaussian sits on top of the other one. Both 2D Gaussians had mean $(10, 10)$, and one had a variance of $(1.0, 1.0)$ while the other had a variance of $(2.0, 2.0)$. This problem does not have an error-less solution, and in fact the best separator, the quadratic where the two Gaussians intersect, has a non-zero Bayes error.

| *technique* | *epochs* | *templates* | *% correct* | *std. dev.* |
|---|---|---|---|---|
| Fuzzy ARTMAP | 8.5 | 220.8 | 63.1 | 1.0 |
| Gaussian ARTMAP ($\gamma = 0.01$) | 5.0 | 12.8 | 66.1 | 10.2 |
| BARTMAP | 9.1 | 32.2 | 68.1 | 2.3 |
| HARTMAP | 3.2 | 54.0 | 69.4 | 1.2 |

Table 1: Generalization Performance – Case 1

In table 1, we see the learning performance of ARTMAP, Gaussian ARTMAP, BARTMAP and HARTMAP

on the problem at hand averaged over 100 sets each consisting of 1000 training samples and 10000 test samples. 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 percentage of correctly classified test instances.

**Overlapping Gaussians–Case 2.** The next pair of distributions considered again has no error-less solution. It consists of two overlapping 2D Gaussians with different means. Thus, the two distributions overlap side-by-side, and a linear boundary is the optimal class separator. One 2D Gaussian had mean $(8, 8)$, and the other one had mean $(12, 12)$, while both had a variance of 2.0.

| *technique* | *epochs* | *templates* | *% correct* | *std. dev.* |
|---|---|---|---|---|
| Fuzzy ARTMAP | 10.7 | 81.3 | 86.7 | 0.9 |
| Gaussian ARTMAP ($\gamma = 0.1$) | 5.0 | 6.0 | 89.0 | 12.8 |
| BARTMAP | 17.4 | 24.8 | 89.6 | 0.7 |
| HARTMAP | 3.2 | 93.6 | 89.3 | 0.9 |

Table 2: Generalization Performance – Case 2

**Overlapping Uniforms–Case 3.** Our third experiment is similar to the first problem, except that now we are dealing with uniform distributions. Both uniform squares have centers at $(10, 10)$. One class has an area of 4, and the other has an area of 16. Note that the Bayes error for this problem is 0.125 %.

| *technique* | *epochs* | *templates* | *% correct* | *std. dev.* |
|---|---|---|---|---|
| Fuzzy ARTMAP | 7.8 | 128.1 | 77.8 | 0.9 |
| Gaussian ARTMAP ($\gamma = 0.1$) | 5.0 | 10.3 | 76.9 | 11.8 |
| BARTMAP | 38.8 | 59.1 | 78.9 | 1.6 |
| HARTMAP | 3.2 | 188.6 | 79.2 | 1.2 |

Table 3: Generalization Performance – Case 3

**Overlapping Bimodal 2D Gaussians–Case 4.** Our next experiment is similar to the second learning problem, except that now we are dealing with bimodal 2D Gaussian distributions. The first class has Gaussian modes with centers at $(1, 1)$ and $(-1, -1)$ each with variances of 1, and the second class has Gaussian modes with centers at $(0, 0)$ and $(0.5, 0.5)$ each with variances of 0.5.

| *technique* | *epochs* | *templates* | *% correct* | *std. dev.* |
|---|---|---|---|---|
| Fuzzy ARTMAP | 8.4 | 163.4 | 72.2 | 1.7 |
| Gaussian ARTMAP ($\gamma = 0.01$) | 5.0 | 12.5 | 75.5 | 12.2 |
| BARTMAP | 9.5 | 45.3 | 75.9 | 2.6 |
| HARTMAP | 3.1 | 57.0 | 77.6 | 1.7 |

Table 4: Generalization Performance – Case 4

**BUPA Liver Disorder.** Performance on the BUPA liver disorder learning problem from the UCI repository is shown in table 5.

| *technique* | *epochs* | *templates* | *% correct* | *std. dev.* |
|---|---|---|---|---|
| Fuzzy ARTMAP | 5.3 | 16.9 | 56.5 | 4.3 |
| Gaussian ARTMAP ($\gamma = 0.1$) | 5.0 | 8.5 | 54.4 | 7.2 |
| BARTMAP | 2.8 | 5.9 | 56.5 | 3.3 |
| HARTMAP | 2.2 | 2.1 | 57.6 | 0.7 |

Table 5: Generalization Performance – BUPA Liver Disorder

# 5 Conclusions

It is clear that HARTMAP has a generalization performance competitive with Fuzzy ARTMAP, Gaussian ARTMAP and BARTMAP on the problems presented. HARTMAP can require a considerable number of $F_2$ nodes compared with the other ART-based neural architectures; however, these extra $F_2$ nodes contain information on the hierarchical structure of the data used in training. Our current research focuses on using the structure contained in the hierarchy for the difficult problem of combined compression and classification.

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

[4] 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.

[5] 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.

[6] 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.

[7] John S. Baras and Subhrakanti Dey, "Combined compression and classification with learning vector quantization," *IEEE Transactions on Information Theory*, vol. 45, no. 6, pp. 1911–1920, 1999.

[8] Catherine L. Blake and C.J. Merz, "UCI repository of machine learning databases [http://www.ics.uci.edu/~mlearn/mlrepository.html]," University of California, Irvine, Dept. of Information and Computer Sciences, 1998.