Safe-μARTMAP: A new solution for reducing category proliferation in Fuzzy ARTMAP

E. Gómez-Sánchez, Y.A. Dimitriadis

University of Valladolid
Dept. T.S.C.I.T.
Camino Viejo del Cementerio
47011 Valladolid, Spain
edugom@tel.uva.es, yannis@tel.uva.es

J.M. Cano-Izquierdo, J. López-Coronado

Polytechnic University of Cartagena
Dept. I.S.A.
Paseo Alfonso XIII 48
30203 Cartagena, MU, Spain
josem.cano@upct.es, jl.coronado@upct.es

Abstract

μARTMAP is a neural network architecture that addresses the category proliferation problem present in Fuzzy ARTMAP, by encouraging the creation of large hyperboxes. However, under certain characteristics of the classification task, this principle can be inadequate, namely if some class has its patterns distributed in several isolated regions, far apart in the input space. Here we propose Safe-μARTMAP, a generalization of μARTMAP that limits the growth of a category in response to a single pattern, so that large hyperboxes are not created under these conditions. Experimental results confirm that performance improves in some synthetic and real world tasks.

1 Introduction

Fuzzy ARTMAP [1] is a neural network architecture capable of establishing an arbitrary mapping in response to collections of arbitrarily complex analog input and output patterns. This is achieved by creating self-organized categories in the input and output spaces, and learning the relations between them. It solves the *stability-plasticity* dilemma, i.e. how to keep on learning without forgetting the knowledge already acquired, and thus overcomes *catastrophic forgetting* happening to other popular networks such as back-propagation trained MLPs. Moreover, Fuzzy ARTMAP has been shown to solve a number of classification tasks in a fast and efficient way [1].

One of the advantages of Fuzzy ARTMAP is that its weights can be easily translated into IF-THEN rules [2], as oppossed to MLPs weights. Nevertheless, the number of categories created during Fuzzy ARTMAP training is usually large, thus yielding an intractable collection of IF-THEN rules. This problem is known as *category proliferation* [1, 2, 9, 6], and has been related to several limitations of Fuzzy ARTMAP, such as the representative inefficiency of hyperbox categories or the excessive triggering of the *match tracking* mechanism due to noisy training patterns [9]. It is also known that Fuzzy ARTMAP depends strongly on the order of presentation of the training patterns, so that particular orders will produce a larger number of categories.

Among the several approaches followed to deal with category proliferation, μARTMAP [5, 6] tries to reduce the number of commited categories by allowing them to be as large as possible, restricted to achieve a certain accuracy in the mapping. However, this approach can be inappropiate if some class has its patternd distributed in several regions in the intput space. Under these conditions, the creation of large categories that link them will also imply an increase of the overlap between categories, and presumably a decrease in accuracy. As a result, μARTMAP will devote more training epochs to reduce this effect, generally creating small unnecessary categories. This is quite noticeably if there are some outliers in the training data, since μARTMAP will enlarge the categories to cover the outliers. Moreover, if the input space is scarcely sampled, creating large categories implies a significant and arbitrary generalization, i.e. it is inferred that patterns in certain regions of the input space are mapped to certain classes without any evidence from the training data. This paper discusses a variation of μARTMAP, called Safe-μARTMAP, that still tries to create large categories, but checking if all the patterns that they cover are close enough, i.e. not allowing large categories to jointly describe all patterns related to one class if they are due to distant sources in the input space.

0-7803-7044-9/01/$10.00 ©2001 IEEE

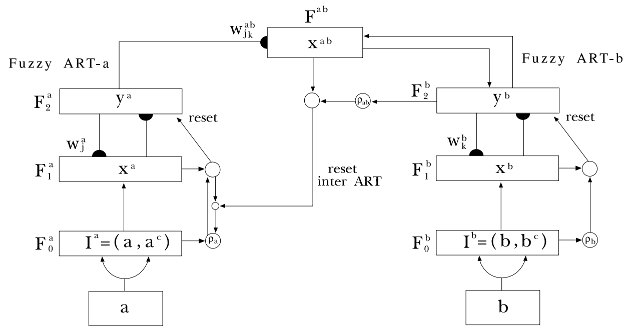

Figure 1: Fuzzy ARTMAP architecture

The rest of this paper is organized as follows: Sections 2 and 3 will briefly recall Fuzzy ARTMAP basics, and the modifications introduced by μARTMAP, respectively. In section 4 Safe-μARTMAP will be presented and discussed. The validity of this approach will be assesed on several benchmarks in section 5. Finally section 6 will draw the main conclusions and future research lines.

2 Fuzzy ARTMAP

Fuzzy ARTMAP architecture, shown in Fig. 1, consists of two Fuzzy ART modules linked through an Inter-ART map. The Fuzzy ART modules perform the self-organized construction of the categories in the input and output spaces, but if the learning task is pattern classification, the ART^b module is not necessary. The ART^a module consists of three layers: input layer F_0, matching layer F_1 and choice layer F_2. In F_0 the input vector $\mathbf{a}$ is normalized by means of complementary code, to yield $\mathbf{I} = (\mathbf{a}, \mathbf{a}^c)$, where $\mathbf{a}^c = \mathbf{1} - \mathbf{a}$. In the choice layer F_2 the input pattern is compared to existing templates, calculating for each template j a choice function $T_j = |\mathbf{w}_j \wedge \mathbf{I}|/(\alpha + |\mathbf{w}_j|)$, where $|\cdot|$ denotes the L^1 norm, and $\wedge$ is the fuzzy intersection or `min` operator. Since the category J with highest T_j is selected to win the competition, the choice parameter α is used to favour the selection of small (especific) categories over large (general) categories [3]. Once a template wins, its degree of matchning to the pattern needs to be sufficient to meet the match condition in F_1, given by $|\mathbf{w}_J \wedge \mathbf{I}|/|\mathbf{I}| \geq \rho$, where ρ is the vigilance parameter. If this condition is not met, category J is inhibitted and another template is selected, or a new one commited.

The selection of a winner category J in ART^a implies a class prediction through the inter-ART map, unless the category has just been commited. If this prediction does not match the correct output class, the *match tracking* mechanism takes place, by both inhibitting category J and increasing the value of the vigilance parameter slightly above $|\mathbf{w}_J \wedge \mathbf{I}|/|\mathbf{I}|$, so that when another category is selected it will be smaller (more especific) than category J.

Once a category J has been selected in ART^a that meets the vigilance criterion and predicts the correct output class, the network enters *resonance*, and weights are updated, by $\mathbf{w}_J^{new} = \beta(\mathbf{w}_J^{old} \wedge \mathbf{I}) + (1 - \beta)\mathbf{w}_J^{old}$, though commonly $\beta = 1$ (then *fast learning* is said to take place).

Because of the use of complementary coding, and of the $\wedge$ operator, if we denote $\mathbf{w}_j = (\mathbf{u}_j, \mathbf{v}_j^c)$, then weights $\mathbf{w}_j$ represent a hyperbox R_j in the input space, where $\mathbf{u}_j$ and $\mathbf{v}_j$ are the lower and upper corners, respectively. Furthermore, if fast learning is used, the hyperbox R_j contains all the training patterns that selected category j during learning.

It is worth pointing out that Fuzzy ARTMAP can be trained either under *incremental* (on-line) or *batch* (off-line) modes. If incremental learning is performed, patterns are presented as they become available, and only once. On the contrary, the batch training consists in repeatedly presenting a collection of patterns until the weights converge. Under *fast learning* conditions it has been proven that training will be over in a finite, generally small, number of list presentations (*epochs*) [7, 4]. Thus, though incremental learning can be important for certain applications, batch training can improve performance at a small computational overhead. Moreover, batch training can be performed initially on a collection of stored data, and weights adapted afterwards with fresh patterns using the incremental learning mode. This paper will only consider the batch training mode of Fuzzy ARTMAP.

3 μARTMAP

The match tracking mechanism allows Fuzzy ARTMAP to detect novelties and learn them correctly, but in the presence of noise causes the creation of many small categories that do not necessarily improve the performance. Moreover, the use of hyperboxes as the descriptive elements associated to categories can be inefficient because they often imply the inference of data on their corners, and more categories may be required to correct the prediction of these corners [9]. However, hyperboxes have some appealing features, as they impose small computational requirements, can facilitate convergence and ease the extraction of knowledge as IF-THEN rules. Moreover, their number can be significantly reduced by setting their possition and size accordingly to the underlying data distribution.

μARTMAP [5, 6] tries to reduce category proliferation by supressing the match tracking mechanism, and creating large categories. Of course, this change implies that some other means need to be provided to guarantee accuracy. In μARTMAP, an inter-ART reset signal is preserved from the original Fuzzy ARTMAP architecture, i.e. nodes in ART^a can be inhibitted by a signal comming from the inter-ART map, but the vigilance parameter ρ will not be raised. In addition, a batch training is carried out, with an evaluation at the end of each iteration of the precission achieved on the training data, as to determine if further epochs are required with more restrivtive vigilance values.

Though complete details can be found in [6], μARTMAP algorithm is briefly described here. μARTMAP introduces a vigilance parameter ρ_j for each category in ART^a, and replaces the inter-ART map by a probabilistic map given by weights p_{jk} and P_{jk}, that link unit j in ART^a and unit k in ART^b, and weights $p_j = \sum_{k=1}^{N^b} p_{jk}$ and $P_j = \sum_{k=1}^{N^b} P_{jk}$ that reflect the probabilistic importance of unit j. All are initialized to 0. User parameters h_{max}, H_{max} and $\Delta\rho$ must be provided. Then the training proceeds in repeated iterations trhough the following two steps.

- **Step 1.** Patterns in the training set are presented to the network. For each of them, winners j and k are selected in ART^a and ART^b, and if some unit j becomes newly committed, $\rho_j = \bar{\rho}$. After winners selection, p_{jk} are tentatively updated to reflect this association, the quatity $h_j = -p_j \sum_{k=1}^{N^b} p_{jk} \log_2 p_{jk}$ is computed and compared to h_{max}. This quantity reflects the homogeneity of the classes associated to unit j. If $h_{max} = 0$, as considered in this paper, each ART^a category can only be related to one output class.

  If $h_j > h_{max}$, then unit j is inhibited, and changes in p_{jk} undone, but no vigilance value is increased. This is what is called an inter-ART reset without match tracking. If $h_j < h_{max}$ then $\mathbf{w}_j$ weights can be updated according to Fuzzy ART laws.

- **Step 2.** Patterns in the training set are presented again, but this time weights are not allowed to change. This is done to evaulate the behaviour of the net on the training data. For each pattern, winners j and k are selected in ART^a and ART^b, that allow to update weights P_{jk}. After all patterns have been presented, the quatity $H = -\sum_{j=1}^{N^a} P_j \sum_{k=1}^{N^b} P_{jk} \log_2 P_{jk}$ is computed and compared to H_{max}.

  If $H > H_{max}$, then the unit J with maximal contribution to H is removed, and all the patterns that selected this unit in the last list presentation are marked for being presented again. This will give place to the creation of new categories, and to assure they will be smaller $\bar{\rho} = |\mathbf{w}_J \wedge \mathbf{I}|/|\mathbf{I}| + \Delta\rho$.

Due to the presence of an inter-ART mechanism, μARTMAP will be able to create hyperboxes inside other hyperboxes, treating correctly the populated exceptions [6] (sets of patterns that cannot be probabilistically ignored, and can be better seen as an *exception rule* to a *general rule*). Moreover, it can allow some training error, avoiding the commitment of small categories to correct the excess of generalization due to hyperboxes near class boundaries, especially if they are curve boundaries, or if classes overlap.

4 Safe-μARTMAP

Though μARTMAP can be a valid solution to reduce category proliferation, as shown for a number of benchmarks in [5, 6], it can be inappropiate under certain conditions. If some class is distributed among several regions, i.e. if its patterns are generated by several distant and not connected sources in the input space, trying to create large categories will complicate the learning and degrade performance. Consider the task shown in Fig. 2a, where the "white" class is clearly disperse, since its patterns can be located in 25 isolated regions. If μARTMAP tries to create large categories that join some of these "white" regions, it will indirectly cause an inaccurate classification of "black" patterns. Therefore, more training epochs will be undergone, in a computationally demanding process, and eventually the categories created will be smaller than required, as shown in figure 2c.

This scenario can also occur because outliers, e.g. rare patterns belonging to a certain class but far from all other patterns of the same class. However, since outliers are probabilistically unimportant, this could be solved by μARTMAP with $h_{max} > 0$.

Safe-μARTMAP is a modification of μARTMAP, that could also be extended to Fuzzy ARTMAP in batch

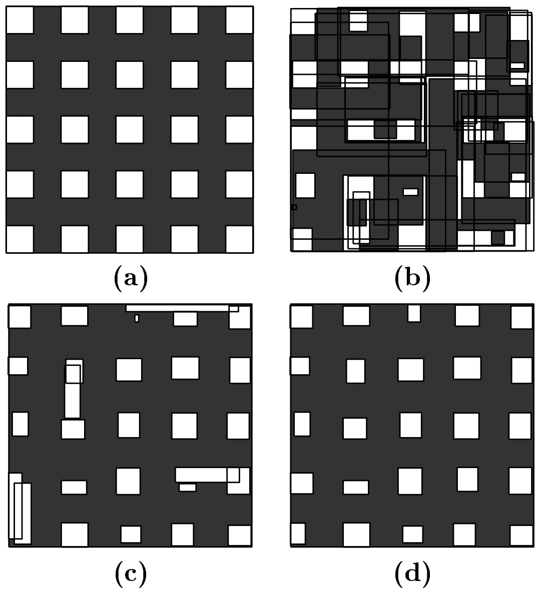

Figure 2: **(a)** A classification task with one class disperse in the input space, and the solutions with fewest categories achieved by **(b)** Fuzzy ARTMAP, **(c)** μARTMAP and **(d)** Safe-μARTMAP

training mode, that will permit the creation of large categories except when they are to include patterns from distant and unconnected sources in the input space, even if they belong to the same output class. To achieve this objective, after a winner category j has been selected in ART^a that meets the vigilance criterion, and it has been checked that its prediction is accepted by the inter-ART map (i.e. $h_j \leq h_{max}$), then the *distance criterion* is evaluated, imposing that if

$$\frac{|\mathbf{w}_j| - |\mathbf{w}_j \wedge \mathbf{I}|}{|\mathbf{I}|} > \delta \qquad (1)$$

where $0 < \delta < 1 - \rho$, then the pattern is too far from the current possition of the category and should not be learned by it. It can be shown that, as well as the total category size is bounded above by $M(1-\rho)$, the change in category size due to a single pattern is bounded above by $M\delta$.

Though a pattern may not meet the distance criterion, the category j may eventually grow to a point where equation (1) would be satisfied. Therefore, it seems convenient to leave the network weights unchanged and save this pattern for future presentation. Therefore, several passes through the training pairs are required to complete a single epoch (considering an epoch one list presentation in which all pairs are actually learned by the network). In successive passes more patterns will be learned, eventually all of them. But it may well occur that some patterns never meet equation (1) with any of the existing categories, $j = 1, \ldots, N^a$. This is because they are far apart from existing categories associated to the same class, and in this case, a new category can be commited by a especial signal. Therefore, large categories will be created if all the training patterns associated to them are somewhat connected. On the contrary, several categories will cover patterns generated by clearly separated sources.

To summarize, letting $\mathcal{P}$ denote the set of training pairs remaining to be presented to the network, the whole training process can be described as follows:

- **Step 1.** Initialize $\mathcal{P}$ to the whole collection of input/output pairs
  - **Step 1a.** For each of the patterns in $\mathcal{P}$, present it to the network. It can either be learned by the network or remain in $\mathcal{P}$.
  - **Step 1b.** If $\mathcal{P}$ is empty, then the epoch is complete. Move to **Step 2**.
  - **Step 1c.** If $\mathcal{P}$ is not empty, and the number of pairs decreased in this pass through data, some patterns have been learned. Then some categories may have changed and thus some of the remaining patterns could be learned in another pass. Go to **Step 1a**.
  - **Step 1d.** If $\mathcal{P}$ is not empty, but the number of pairs in it is the same as in previous pass, then inhibitt all categories and present one pattern from $\mathcal{P}$. This forces the creation of a new category. For the remaining patterns in $\mathcal{P}$ go to **Step 1a**.
- **Step 2.** Evaluate the total entropy H as described above for μARTMAP, and if necessary go again to **Step 1**.

It is worth mentioning that if $\delta = 1$ any pattern can be learned by any category, and thus Safe-μARTMAP reduces to μARTMAP.

5 Experimental results

In order to assess the performance of the different architectures, several benchmarks are considered. First, Fig. 2 shows a synthetic benchmark in which one of the classes is very disperse. This will cause a degradation in μARTMAP's performance, as discussed above. In addition, the *circle-in-the-square* problem [1], commonly

Table 1: Training and generalization results for the classification task shown in Figure 2, averaged over 10 simulations

Architecture	#Cat	Error
Fuzzy ARTMAP	62.0	14.03%
μARTMAP	41.1	13.29%
Safe-μARTMAP	26.0	6.74%

used in the ART literature, will be used to ilustrate how outliers can affect μARTMAP's performance, and how this effect can be differently overcome by an adequate selection of h_{max} or by the use of Safe-μARTMAP. Finally, the architectures under study will be evaluated on a real world problem from the UCI machine learning repository [8], the Pima Indians Diabetes (**PID**) problem, that features high input dimensionality and few data points for training, so that creating large categories will be possible, but at the risk of generalizing in excess.

In all experiments the results shown are the average of ten simulations, with different training data. For the synthetic benchmarks we generated ten 1,000-point training sets and one 10,000-point test set, from random sources with uniform distribution in $[0,1] \times [0,1]$. For the PID database, the 768 available patterns were split randomly into 576-point training sets and 192-point test sets, as in [2], and this was done ten times for the different simulations. In all networks $\rho = 0$ so that the category size was only constrained by the data, $\beta = 1$ (fast learning), $\alpha \cong 0$ and $\Delta\rho = 0.01$. Other user parameters were selected by cross-validation on different data.

The results achieved for the task in Fig. 2 are shown in Table 5, using $h_{max} = 0$ and $H_{max} = 0.1$ and $\delta = 0.05$. In addition, Fig. 2 shows the *best* result (in terms of fewest categories) achieved by each of the networks. This figure illustrates why the fact that one class has its patterns distributed in several regions affects both Fuzzy ARTMAP and μARTMAP, that tend to join these patterns in one single category. While Fuzzy ARTMAP solves this creating more, smaller categories, μARTMAP repeatedly passes trhough the data, destroying and creating new categories to divide the different regions, but as a lateral result creating smaller categories. On the contrary, Safe-μARTMAP can isolate these regions by evaluating the distance criterion. Of course, this is an extremely favourable case for Safe-μARTMAP. In general, Safe-μARTMAP will have to combine the use of this criterion with the evaluation of entropy inherited from μARTMAP, in order to solve real problems successfully.

Table 2: Training and generalization results for the circle-in-the-square problem, without and with noisy samples, averaged over 10 simulations

Outliers	Architecture	#Cat	Error
No	Fuzzy ARTMAP	25.2	5.70%
	μARTMAP	9.0	5.64%
	Safe-μARTMAP	9.0	5.64%
Yes	Fuzzy ARTMAP	37.0	7.14%
	μARTMAP$_{h_{max}=0}$	19.6	7.16%
	μARTMAP$_{h_{max}=0.025}$	29.2	5.90%
	Safe-μARTMAP	18.5	5.87%

To study the influence of outliers, as a particular cause of dispersion among patterns belonging to on class, the *circle-in-the-square* problem has been considered. This problem consists in determining wether points inside the unit square $[0,1] \times [0,1]$ lie within or outside a circle of area $\frac{1}{2}$ centered in $[\frac{1}{2}, \frac{1}{2}]$. In [6] we showed that, on clean data, μARTMAP clearly outperforms Fuzzy ARTMAP, achieving smaller error rate with far fewer categories. These results (with $h_{max} = 0$, $H_{max} = 0.18$) are also presented in table 5, where $\delta = 1$ for Safe-μARTMAP, so that in fact behaves as μARTMAP. However, 10 outliers were introduced in the training data (representing 1% of them), and μARTMAP performance clearly degrades. This is due to the fact that it tries to create large categories that link the circle to the outliers, but these categories overlap with the "outside" class, so that several epochs required to solve this problem, and some smaller categories become commited.

However, Safe-μARTMAP can overcome this difficulty by creating special categories for the outliers, that are far from the circle, and keeping the rest the same, so that a larger number of categories is generated but the performance is preserved. These data can also be handled by increasing parameter h_{max}, so that during learning ART^a may be mapped to more than one output class. Nevertheless, though this change allows handling the outliers, it also causes some categories to grow in response to patterns with an incorrect class label. Namely, the categories describing the class "circle" can learn some of the patterns nearby, though they should be mapped to the class "outside", and then further refinement will be required. In summary, precission can be kept at the cost of creating some smaller categories, as also shown in Table 5 for μARTMAP trained with $h_{max} = 0.025$.

Table 3: Training and generalization results for the PID database, averaged over 10 simulations

Architecture	#Cat	Error
Fuzzy ARTMAP	48.30	32.08%
μARTMAP	15.10	32.03%
Safe-μARTMAP	27.50	29.48%

To complete the experimental assessment of the proposed methods, the Pima Indians Diabetes database has been selected from the machine learning repository held at UCI [8]. It consists of the data of 768 pacients, each characterized by 8 numeric valued features, that should allow to determine wether a pacient develops diabetes. As the dimensionality of the problem is high, it is likely that patterns corresponding to the same class lie far apart in the input space. However, since data is scarce, it is quite likely that these large hyperboxes are allowed. As they are used to classify unseen data they can make risky generalizations, and then performance degrades. The results achieved with $h_{max} = 0$, $H_{max} = 0.6$ and $\delta = 0.09$, are shown in Table 5. The intrinsic difficulty of this problem is clear from the high error rates achieved by all networks. However, it is clear that μARTMAP can achieve significant code compression, by placing large categories in the input space. Unfortunately, the consequence of this is that performance degrades, as already discussed. This suspect is confirmed by the fact that Safe-μARTMAP creates a few more categories reducing the error rate in 2.5%. Finally, we should point out that though performances reported in [2] are better, even for Fuzzy ARTMAP, in [2] predictions are computed after voting acrosss 20 networks, so actually the categories (rules) contributing to the prediction are 20 times as many.

6 Conclusions and future research

This paper has presented a modification of μARTMAP, called Safe-μARTMAP, that can enhance its performance in particular scenarios, namely if some classes are distributed in distant regions in the input space, especially if dimmensionality is high. The algorithm encourages the creation of large categories, in order to reduce their number, but does not let them cover patterns corresponding to isolated regions, distant in the input space, even they are related to the same output class. This is achieved by evaluating the distance criterion just before learning ocurrs, so that hyperboxes are only allowed to enlarge in response to close patterns. Experimental results have shown the usefulness of this modification in some synthetic and real world classification tasks, while setting $\delta = 1$ can reduce Safe-μARTMAP to μARTMAP when ceonvenient.

These ideas could also be extended to Fuzzy ARTMAP. However, it can be expected that its generalization performance will not be significantly affected, because the match tracking mechanism helps to handle disperse classes correctly, though generally at the cost of creating many small categories. Despite of that, the evaluation of the distance criterion causes a kind of ordering in the presentation of the training patterns and in the way categories are commited: existing categories grow slowly as they learn patterns nearby. This will probably reduce the number of times the match tracking mechanism is triggered, and consequently we may expect the number of commited categories to reduce.

References

[1] G.A. Carpenter, S. Grossberg, N. Markuzon, J. Reynolds, and D.B. Rosen. Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks*, 3(4):698–713, September 1992.

[2] G.A. Carpenter and H.A. Tan. Rule extraction: From neural architecture to symbolic representation. *Connection Science*, 7(1):3–27, 1995.

[3] M. Georgiopoulos, H. Fernlund, G. Bebis, and G. Heileman. Order of search in Fuzzy ART and Fuzzy ARTMAP: Effect of the choice parameter. *Neural Networks*, 9(9):1541–1559, 1996.

[4] M. Georgiopoulos, J. Huang, and G. Heileman. Properties of learning in ARTMAP. *Neural Networks*, 7(3):495–506, 1994.

[5] E. Gómez-Sánchez, Y.A. Dimitriadis, J.M. Cano-Izquierdo, and J. López-Coronado. MicroARTMAP: use of mutual information for category reduction in Fuzzy ARTMAP. In *Proceedings of the International Joint Conference on Neural Networks, IJCNN2000*, volume VI, pages 47–52, Como, Italy, July 2000.

[6] E. Gómez-Sánchez, Y.A. Dimitriadis, J.M. Cano-Izquierdo, and J. López-Coronado. μARTMAP: use of mutual information for category reduction in Fuzzy ARTMAP. To appear in *IEEE Transactions on Neural Networks*, 2001.

[7] J. Huang, M. Georgiopoulos, and G. Heileman. Fuzzy ART properties. *Neural Networks*, 8(2):203–212, 1995.

[8] P. Murphy and D. Ada. UCI repository of machine learning databases. Technical report, Department of Computer Science, University of California, Irvine, California, 1994.

[9] J. Williamson. Gaussian ARTMAP: A neural network for fast incremental learning of noisy multidimensional maps. *Neural Networks*, 9(5):881–897, 1996.