VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wananga o te Upoko o te Ika a Maui

Department of Computer Science

PO Box 600
Wellington
New Zealand

Tel: +64 4 471 5328
Fax: +64 4 495 5232
Internet: Tech.Reports@comp.vuw.ac.nz

An Improved Learning Algorithm for the Fuzzy ARTMAP Neural Network

G. Bartfai

Technical Report CS-TR-95/10
July 1995

Abstract

This article introduces two improvements to the learning algorithm of the Fuzzy ARTMAP neural network. One of them is concerned with the timing according to which input patterns and their corresponding target output are processed by the network. The other one is the explicit overwriting of an existing association between an input and an output category in case the input is matched perfectly and yet the network's prediction is wrong. Both of these modifications are needed to reduce the occurrence of the "match tracking anomaly" (or *MTA*) during learning, and eliminate *MTA* altogether in a trained network. As a result, training time is also reduced, which is demonstrated through the performance of the network on a machine learning benchmark database.

Publishing Information

A shortened version of this report is to appear in *Proceedings of ANNES'95, The Second New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems, Dunedin, New Zealand*, published by IEEE Computer Society Press.

Author Information

Guszti Bartfai is with the Department of Computer Science.
His e-mail address is guszti@comp.vuw.ac.nz

1 Introduction

The *Fuzzy ARTMAP* neural network [5] is a supervised learning mechanism capable of self-organising stable recognition categories in response to arbitrary sequences of analogue and binary input patterns. It offers a unique solution to the *stability-plasticity dilemma* that autonomous learning systems have to face, and overcomes the problems of long training and "catastrophic forgetting" associated with many popular networks such as backpropagation [9]. The Fuzzy ARTMAP network has been applied to several machine learning benchmark problems [5] as well as real-world tasks (for example [1]), and compared favourably with other neural network or traditional AI learning methods.

There are, however, anomalous situations that can occur — under certain conditions — during training as a result of its internal "match tracking" process. This was first reported in [6] for the ARTMAP network, but was also claimed that it would never arise if the (binary) input patterns have the same number of 1's [6, pages 584-585]. We showed in [3] that this claim holds only if the *vigilance* parameter of the ART_b module (ρ_b, see in section 2) is set to 1. Although $\rho_b = 1$ is a typical setting in pattern recognition, there can be applications where $\rho_b < 1$ is needed (see [2] for example).

This paper looks at the Fuzzy ARTMAP network, and introduces two improvements to its learning algorithm: one that eliminates the *match tracking anomaly* (or *MTA*) when the network is in fast learning mode, and another that is needed to initially correct and eventually eliminate the *MTA* — on a finite training set — in slow learning mode. The effect of these improvements are demonstrated when the network is trained on a machine learning benchmark database.

Section 2 introduces the Fuzzy ARTMAP network at a level that is necessary for understanding the main results of this paper. Section 3 deals with the Match Tracking Anomaly (*MTA*) in the Fuzzy ARTMAP network. In section 4, the two improvements to the standard Fuzzy ARTMAP learning algorithm are described. The experiments are discussed and the results are presented in section 5. Finally, conclusions are drawn in section 6.

2 The Fuzzy ARTMAP network

The Fuzzy ARTMAP neural network architecture is capable of incremental learning of multidimensional mappings in response to arbitrary sequences of analogue or binary input/target vector pairs [5]. The vectors can represent fuzzy or crisp sets of features. The architecture of the network (see figure 1) is identical to that of an ARTMAP network [6], but the binary ART1 [4] modules are replaced with Fuzzy ART [7] ones.

A Fuzzy ART network performs unsupervised clustering on analogue and binary input patterns. It has three layers: the *input* layer ($F0$), the *comparison* layer ($F1$), and the *recognition* layer ($F2$), with M, M and N neurons, respectively (see modules ART_a and ART_b in figure 1). The neurons, or nodes, in the F2 layer represent input categories. Each F2 node j stores a prototype vector $\mathbf{z}_j$ for the category it represents.

Upon presentation of an input vector $\mathbf{x}$ ($x_i \in [0,1], i = 1,2,\ldots,M$), the network attempts to classify it into one of its existing categories based on its similarity to the stored prototype of each category node. More precisely, for each node j in the F2 layer, the *choice function*

$$T_j(\mathbf{x}) = \frac{|\mathbf{x} \wedge \mathbf{z}_j|}{\beta + |\mathbf{z}_j|} \tag{1}$$

is calculated, where $\beta > 0$ is the *choice parameter*, the fuzzy AND [10] operator $\wedge$ is defined by

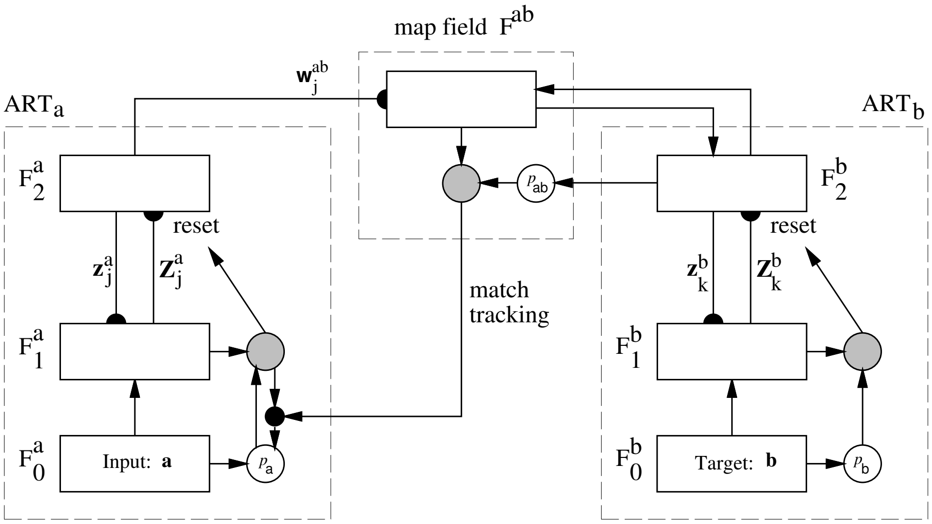

Figure 1: Architecture of the Fuzzy ARTMAP network. (This figure is a simplified version of the one that appeared in [5].) Note that the bottom-up weights $\mathbf{Z}_j^a$ and $\mathbf{Z}_k^b$ are not used explicitly in Fuzzy ART, but are included in (1) implicitly.

$$(\mathbf{p} \wedge \mathbf{q})_i \equiv min(p_i, q_i) \quad ,$$

and the norm $|\cdot|$ is defined by

$$|\mathbf{p}| \equiv \sum_{i=1}^{M} p_i$$

for any M-dimensional vector $\mathbf{p}$.

The network then makes a category choice by selecting the F2 node J where

$$T_J = \max\{T_j : j = 1, 2, \ldots, N\} \ .$$

The F1 layer activity vector then becomes $\mathbf{x} \wedge \mathbf{z}_J$ and is compared to the current input vector $\mathbf{x}$: the strength of the match is given by

$$\frac{|\mathbf{x} \wedge \mathbf{z}_J|}{|\mathbf{x}|} \ , \tag{2}$$

which is compared with a system parameter ρ called *vigilance* $(0 \leq \rho \leq 1)$. If the input matches sufficiently, i.e. the match strength $\geq \rho$, then it is assigned to F2 node J and the network learns it by adjusting $\mathbf{z}_J$ according to the equation

$$\mathbf{z}_J^{(new)} = \eta(\mathbf{x} \wedge \mathbf{z}_J^{(old)}) + (1 - \eta)\mathbf{z}_J^{(old)} \tag{3}$$

where η is the *learning rate* $(0 \leq \eta \leq 1)$.

If the stored prototype $\mathbf{z}_J$ does not match the input sufficiently (match strength $< \rho$), the winning F2 node J is reset for the period of presentation of the current input. Then another F2 node (or category) will be selected whose prototype will be matched against the input. This "hypothesis-testing" cycle is repeated until the network either finds a stored category whose prototype matches the input closely enough, or allocates a new F2 node. Then learning takes place as described above.

After an initial period of self-stabilisation, the network will directly (i.e. without search) access the prototype of one of the categories it has found in a given training set. The higher the vigilance level, the larger number of smaller, or more specific, categories will be created. (If $\rho = 1$, the network will assign a new category for every unique input pattern.)

The two Fuzzy ART modules in a Fuzzy ARTMAP network are linked together by an "inter-ART" associative memory called *map field* (F^{ab}). Module ART$_a$ (with a *baseline* vigilance $\overline{\rho_a}$) learns to categorise input patterns $\mathbf{a}^{(i)}$ presented at layer F_0^a, while module ART$_b$ (with vigilance ρ_b) develops categories of target patterns $\mathbf{b}^{(i)}$ presented at layer F_0^b. Modules F_2^a and F^{ab} are connected via associative links (w_{jk}^{ab}) whose strengths are adjusted through learning (see below). There are one-to-one, two-way, and non-modifiable connections between nodes in the F^{ab} and F_2^b layers, i.e. each F_2^b node is connected to its corresponding F^{ab} node, and vice versa. A new association between an ART$_a$ category J and an ART$_b$ category K is learned by setting the corresponding $F_2^a \rightarrow F^{ab}$ link (or connection weight) to one and all other links from the same ART$_a$ node to zero, i.e.

$$w_{jk}^{ab} = \begin{cases} 1 & \text{if } j = J \text{ and } k = K \\ 0 & \text{otherwise} \quad . \end{cases} \tag{4}$$

When an input pattern is presented to the network, the F^{ab} layer will receive inputs from both the ART$_a$ module ($\mathbf{w}_J^{ab}$ vector through the previously learned $J \rightarrow K$ associative link) and the ART$_b$ module (F_2^b output vector $\mathbf{y}^b$ with the winning F_2^b node set to one). If the two F^{ab} inputs match, i.e. the network's prediction is confirmed by the selected target category, the network will learn by modifying the prototypes of the chosen ART$_a$ and ART$_b$ categories according to the Fuzzy ART learning equations shown above. If there is a mismatch at the F^{ab} layer, a map field reset signal will be generated, and a process called *match tracking* will start whereby the baseline vigilance level of the ART$_a$ module will be raised by the minimal amount needed to cause mismatch with the current ART$_a$ input at the F_1^a layer. This will subsequently trigger a search for another ART$_a$ category, whose prediction will be matched against the current ART$_b$ category at the F^{ab} layer again. This process continues until the the network either finds an ART$_a$ category that predicts the category of the current target correctly, or creates a new F_2^a node and a corresponding link in the map field, which will learn the current input/target pair according to (4). The ART$_a$ vigilance is then allowed to return to its resting level ($\overline{\rho_a}$).

After a few presentations of the entire training set, the network will self-stabilise, and will read out the expected output for each input without search.

3 The Match Tracking Anomaly in the Fuzzy ARTMAP network

The *MTA* in the Fuzzy ARTMAP network occurs when an input category matches the current input pattern perfectly and yet the network predicts a wrong output category. In specific, suppose ART$_a$ input vector $\mathbf{a}^{(i)}$ is perfectly coded by category J, i.e. $\mathbf{z}_J^a = \mathbf{a}^{(i)}$. (This is the case, for example, when category J is created, or node J becomes committed, while input $\mathbf{a}^{(i)}$ is presented.) Furthermore, the network has learned to associate ART$_b$ category K with ART$_a$ category J. Suppose the prototype vector $\mathbf{z}_K^b$ of category K is subsequently recoded during further training such that it no longer matches target $\mathbf{b}^{(i)}$ enough at the current $\rho_b < 1$ vigilance level, but $\mathbf{z}_J^a$ remains unchanged. So when the ($\mathbf{a}^{(i)}$, $\mathbf{b}^{(i)}$) pair is presented again, ART$_b$ will select a different category, which will cause category mismatch at the map field with the network's prediction of category K. This triggers the match tracking process whereby the network will attempt to raise ρ_a above 1 (since prototype $\mathbf{z}_J^a$ matches input $\mathbf{a}^{(i)}$ perfectly). The network will then be unable to find another ART$_a$ category that would match the input better.

In such situations, the network does not learn on the current input/target pair. This would be an adequate solution if the *MTA* could be eliminated by suitable preprocessing of inputs such as *complement coding*, as was suggested in [6].

However, the *MTA can* in fact arise in the Fuzzy ARTMAP network for the following two reasons:

1. Noting that a Fuzzy ARTMAP network in fast learning mode (i.e. $\eta = 1$) is equivalent to ARTMAP if the inputs are binary, the timing conditions (see [3] and below) will be crucial here, too, when $\rho_b < 1$.

2. If the Fuzzy ARTMAP network is trained in slow learning mode (i.e. $\eta < 1$), it can no longer be guaranteed that $\mathbf{z}_K^b = \mathbf{b}^{(i)} \wedge \mathbf{z}_K^b$ once F_2^b node K has encoded target output $\mathbf{b}^{(i)}$ according to (3). This was, however, an important assumption for the ARTMAP network to avoid *MTA* (see theorem 1 in [3]).

4 Improvements to the learning algorithm

We propose the following two improvements to the original Fuzzy ARTMAP learning algorithm (which will be called FAM henceforth).

Improvement 1 ("priming"): Upon presentation of each $(\mathbf{a}^{(i)}, \mathbf{b}^{(i)})$ pair from the training set, find an ART$_a$ category J for the $\mathbf{a}^{(i)}$ input first, select an output category K (via the F^{ab} map field), then "prime" the F_1^b layer with the prototype vector $\mathbf{z}_K^b$ of the selected F_2^b node before presenting the target output $\mathbf{b}^{(i)}$[1]. This is essentially the "timing condition" of theorem 1 in [3] applied here to the Fuzzy ARTMAP network.

Improvement 2 ("unlearning"): Once the *MTA* condition is detected[2], "unlearn" the existing $J \to K$ link and create a new $J \to K'$ one where K' is the selected ART$_b$ category on the current target pattern.

The learning algorithm with the above modifications will be referred to as FAM-UP (for "FAM with Unlearning and Priming").

Note that the above improvements will *not* eliminate the *MTA* altogether (see section 6), but will

- minimise the occurrence of *MTA*, and
- ensure *MTA* will not re-emerge on the same input/target pair from a finite training set.

As a consequence of the reduction in *MTA* occurrences, training time is also reduced, which is demonstrated in the following section.

[1] This way node K will be tested against vigilance before any other F_2^b node, which may not happen were the ART$_b$ module left to categorise $\mathbf{b}^{(i)}$ independently.

[2] Note that it can only occur in slow learning mode if improvement 1 is already applied.

ART_b learning rate η_b	Training time in epochs		*MTA* % of Epochs		*MTA* % of Trials	
	FAM	FAM-UP	FAM	FAM-UP	FAM	FAM-UP
0.1	14.66	10.14	79.60	25.44	4.42	0.33
0.2	11.49	6.18	81.54	31.71	3.86	0.41
0.3	10.45	4.45	73.77	33.25	3.18	0.43
0.4	9.36	3.27	63.46	27.21	2.34	0.36
0.5	9.03	2.90	61.57	22.75	1.89	0.31
0.6	9.64	2.69	60.16	22.67	2.74	0.30
0.7	7.91	2.24	52.46	13.83	2.13	0.17
0.8	6.79	2.76	49.77	9.42	2.13	0.10
0.9	5.03	1.97	29.22	1.01	0.75	0.01
1.0	2.12	1.94	11.32	0	0.19	0

Table 1: This table compares the original and improved Fuzzy ARTMAP learning algorithms when the ART_b learning rate changes from 0.1 to 1. The ART_a learning rate η_a was set to 1. Each row shows averages over the same set of 100 training sessions. In each training session, the network was trained completely on the training set, i.e. training stopped when no further weight changes occurred in the network for 5 full training cycles (or epochs). The ART_a and ART_b vigilance parameters (ρ_a, ρ_b) were 0.4 and 0.1, respectively. The "*MTA* % of epochs" columns show the percentage of epochs in which *MTA* occurred at least once. The "*MTA* % of trials" columns show the percentage of learning trials (i.e. input/target presentations) in which *MTA* occurred during the 100 training sessions. The choice parameters (β_a and β_b) were set to 0.001.

5 Experimental results

In order to evaluate the new Fuzzy ARTMAP learning algorithm (FAM-UP), we carried out experiments on the "zoo" machine learning benchmark database [8]. Unlike in [6], where target outputs were codes for correct output categories, here the Fuzzy ARTMAP network was used in an "auto-associative" setting to develop two-level class hierarchies from the training set (similar to [2]). In particular, each input was identical to its target pair (i.e. $\mathbf{a}^{(i)} \equiv \mathbf{b}^{(i)}$), and both the ART_a and ART_b modules developed clusterings of patterns from the same training set at different levels of vigilance ($\rho_a > \rho_b$).

The "zoo" database contains 101 instances of animals described with 18 attributes. Out of these attributes, we used the 15 boolean ones that indicate the presence or absence of certain features like "hair", "aquatic", "domestic" and so on. We also used the "number of legs" attribute, which is a set of 6 integers. The "type" (or class) attribute was ignored.[3] The patterns were presented to the Fuzzy ARTMAP network in *complement coding* resulting in 36-dimensional binary vectors of equal norm (= 16).

Training proceeded by showing each input/target pair from the training set repeatedly until convergence was reached, i.e. the network stabilised itself on the training set. Convergence was established if the number of input and target patterns for each ART_a and ART_b category did not change for the past 5 presentations of the entire training set. After training, each input and target pattern had direct access, i.e. without search, to its corresponding prototype.

Table 1 shows a comparison of the FAM and FAM-UP algorithms.

[3] The 18th attribute ("animal name") was simply used as a label for the individual instances.

It can be seen from the table that applying FAM-UP reduced training time by 9 to 72%. We can also observe a 2 to 30-fold decrease in the number of epochs where *MTA* occurred during training. A more significant (6 to 75-fold) reduction is shown in the "*MTA* per trial" measure with the improved algorithm. Note that the *MTA* was eliminated completely when the network stabilised itself.

6 Conclusion

In this article, we discussed two improvements to the standard Fuzzy ARTMAP learning algorithm: one that is concerned with the timing at which ART_a inputs and ART_b targets are presented to the network; and another that explicitly "unlearns" an $F_2^a \rightarrow F_2^b$ link once the *MTA* occurs. We have shown that these modifications not only result in faster training, but also reduce the occurrence of *MTA* significantly. Moreover, if the network is trained on a finite training set, the *MTA* can be eliminated this way completely. If, however, the network is used in a real-time autonomous learning environment, the *MTA* cannot be eliminated, but at least can be *detected*. Note that in real-world environments the *MTA* can be caused by contradicting input/target patterns. How the network should handle such situations could be the subject of further investigation.

In the future, we plan to test the new learning algorithm on a variety of machine learning problems as well as analyse the effect of changing vigilance levels (ρ_a, ρ_b) and learning rates (η_a, η_b).

Acknowledgement

This work was in part supported by the Department of Computer Science at Victoria University of Wellington, New Zealand. I wish to thank Andrew Vignaux for his programming support.

References

[1] D. Alpsan. Auditory Evoked Potential Classification by Unsupervised ART 2-A and Supervised Fuzzy ARTMAP Networks. In *Proceedings of the IEEE International Conference on Neural Networks*, volume 6, pages 3512–3515. IEEE Press, 1994.

[2] G. Bartfai. Hierarchical Clustering with ART Neural Networks. In *Proceedings of the IEEE International Conference on Neural Networks*, volume 2, pages 940–944. IEEE Press, 1994.

[3] G. Bartfai. On the Match Tracking Anomaly of the ARTMAP Neural Network. Technical Report CS-TR-95/1, Department of Computer Science, Victoria University of Wellington, New Zealand, January 1995. Accepted for publication in *Neural Networks*.

[4] G.A. Carpenter and S. Grossberg. A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine. *Computer Vision, Graphics, and Image Processing*, 37:54–115, 1987.

[5] G.A. Carpenter, S. Grossberg, N. Markuzon, J.H. 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(5):698–713, September 1992.

[6] G.A. Carpenter, S. Grossberg, and J.H. Reynolds. ARTMAP: Supervised Real-Time Learning and Classification of Nonstationary Data by a Self-Organizing Neural Network. *Neural Networks*, 4:565–588, 1991.

[7] G.A. Carpenter, S. Grossberg, and D.B. Rosen. Fuzzy ART: Fast Stable Learning and Categorization of Analog Patterns by an Adaptive Resonance System. *Neural Networks*, 4:759–771, 1991.

[8] P. M. Murphy and D. W. Aha. UCI Repository of Machine Learning Databases [Machine-readable data repository]. Technical report, Department of Information and Computer Science, University of California, Irvine, CA, 1992.

[9] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning internal representations by error propagation. In D.E. Rumelhart and J.L. McClelland, editors, *Parallel Distributed Processing; Explorations in the Microstructure of Cognition; Foundations*, volume 1, pages 318–362. The MIT Press, 1986.

[10] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.