**Monograph:**
Lim, Chee Peng and Harrison, R.F. (1995) Minimal Error Rate Classification in a Non-stationary Environment via a Modified Fuzzy ARTMAP Network. Research Report. ACSE Research Report 557 . Department of Automatic Control and Systems Engineering

# Minimal Error Rate Classification in a Non-stationary Environment via a Modified Fuzzy ARTMAP Network

Chee Peng Lim and Robert F. Harrison

Department of Automatic Control and Systems Engineering
The University of Sheffield
PO Box 600, Mappin Street
Sheffield, S1 4DU, UK
Electronic mail: c.lim@sheffield.ac.uk, r.f.harrison@sheffield.ac.uk

## Abstract

This paper investigates the feasibility of the fuzzy ARTMAP neural network for statistical classification and learning tasks in an on-line setting. The inability of fuzzy ARTMAP in implementing a one-to-many mapping is explained. Thus, we propose a modification and a frequency measure scheme which tend to minimise the misclassification rates. The performance of the modified network is assessed with noisy pattern sets in both stationary and non-stationary environments. Simulation results demonstrate that modified fuzzy ARTMAP is capable of learning in a changing environment and, at the same time, of producing classification results which asymptotically approach the Bayes optimal limits. The implications of taking time averages, rather than ensemble averages, when calculating performance statistics are also studied.

Research Report No. 557
January 1995

# 1 Introduction

Feedforward neural networks such as the Multilayered Perceptron (MLP) networks and the Radial Basis Function (RBF) networks possess some attractive properties when the objective is to develop a classifier to operate in a probabilistic environment. These network structures have been proven to be able to represent any smooth enough function to an arbitrary degree of accuracy [1,2]. Thus, it is likely that feedforward networks can offer a direct solution to the problem of developing a one-from-many classifier. However, such an approach is only viable when there is good reason to believe that the data environment is stationary and that the data sample used in training is sufficiently representative. In cases where learning takes place in a non-stationary environment, it is either necessary to allow the feedforward networks to carry on learning or to re-train them off-line. Nevertheless, it is well-documented that networks of the Adaptive Resonance family offer a way out of this problem—the so-called stability-plasticity dilemma [3,4]. They are able to learn continuously in a changing data environment and acquire knowledge *in situ* whilst simultaneously providing useful classification results. This paper investigates the classification ability of fuzzy ARTMAP (FAM) [5], a variant of the supervised Adaptive Resonance Theory (ART) networks, in purely statistical learning tasks and compares the results with Bayesian decision theorem.

# 2 Fuzzy ARTMAP (FAM)

FAM is an extension of the ARTMAP network [6] which makes use of the operation of fuzzy set theory instead of the classical set theory that governs the dynamics of ARTMAP. A FAM network consists of two fuzzy ART [7] modules, $ART_a$ and $ART_b$, connected by a map field as shown in Fig. 1. During supervised learning, an input pattern vector $\boldsymbol{a}$ is fed to $ART_a$ with its target vector $\boldsymbol{b}$ to $ART_b$. $ART_a$ and $ART_b$ cluster their input vectors independently. An intervening map field ($F_{ab}$) adaptively associates predictive antecedents in $ART_a$ with their consequents in $ART_b$.

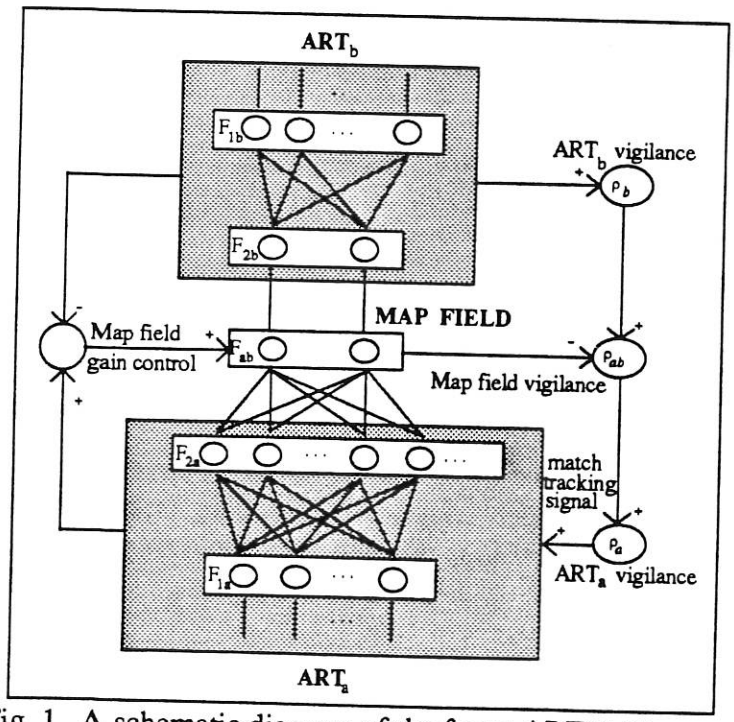


Fig. 1 A schematic diagram of the fuzzy ARTMAP network

In $ART_a$ (as well as in $ART_b$), an input vector $\boldsymbol{a}$ first registers itself at the $F_{1a}$ layer in complement-coded format, *i.e.* $A=(\boldsymbol{a}, 1\text{-}\boldsymbol{a})$, to avoid the category proliferation problem [7,8]. This pattern vector is faned-out to all the nodes in the $F_{2a}$ layer via a set of Long Term Memories (LTMs) or weights. The response of each $F_{2a}$ node is based upon a fuzzy *choice function*

$$\frac{|A \wedge w_{a-j}|}{\alpha_a + |w_{a-j}|}$$

where $w_{a-j}$ is the weight vector of the $j$th $F_{2a}$ node, $\alpha_a$ is the choice parameter [5,7] of $ART_a$ and the fuzzy "and" operator ($\wedge$) and the norm $|.|$ are defined as: $(x \wedge y)_i \equiv \min(x_i, y_i)$ and $|x| = \sum_i |x_i|$ [9]

The maximally activated node is selected as the winner and all other nodes are suppressed in accordance with the winner-take-all competitive structure. The winning $F_{2a}$ node then feeds back its weight vector to $F_{1a}$. This weight vector represents the category prototype of the winning node and is used for comparison with the input vector against a vigilance threshold. Resonance is said to occur if the vigilance test is satisfied, *i.e.*

$$\frac{|A \wedge w_{a-J}|}{|A|} \geq \rho_a$$

where $\rho_a$ is the vigilance parameter of $ART_a$ and $w_{a-J}$ is the winning $J$th node in $F_{2a}$. Otherwise, a mismatch signal is sent to $F_{2a}$ to reset the winning node for the rest of the pattern matching cycle. The input vector $\boldsymbol{A}$ is now re-transmitted to $F_{2a}$ to select a new winner. This search cycle ends when the current category prototype is able to meet the vigilance test or a new node is recruited in $F_{2a}$ with the input pattern coded as the prototypical weight vector.

After resonance has occurred in $ART_a$ and $ART_b$, a predictive signal is sent from the winning $F_{2a}$ node to the map field. If this prediction is disconfirmed by the winning node in $F_{2b}$, *i.e.* the map field vigilance test fails, a control strategy called match-tracking is initiated. Match-tracking increases $\rho_a$ to a value which triggers a search in $ART_a$. Thus, $\rho_a$ is made slightly greater than $|A \wedge w_{a-J}|/|A|$ to cause the $ART_a$ vigilance test to fail. In such a way, match-tracking provides a means to select a node in $F_{2a}$ which fulfils both the $ART_a$ and the map field vigilance tests. If such a node does not exist, $F_{2a}$ is *shut down* for the rest of the input presentation [5,6].

## 2.1 One-to-many Mapping

One-to-many mapping is defined as the formation of an association from an $F_{2a}$ category node to more than one $F_{2b}$ target output via the map field. Obviously, this association is prohibited in the ARTMAP (both the binary and fuzzy versions) networks to avoid any confusion during recall, hence prediction, from an $F_{2a}$ category node to its $F_{2b}$ predictive answer. However, in statistical pattern classification, overlapping regions can occur in the input space where the same cluster may belong to more than one target output, subject to different probabilities of class membership. It is therefore useful if this one-to-many mapping can be established.

## 2.2 Modified Fuzzy ARTMAP

One way to implement the one-to-many mapping has been proposed in [10] and is explained below. When the input vector is a fuzzy subset of an $F_{2a}$ category node, it will match perfectly with the category prototype and the $ART_a$ vigilance test produces $|A \wedge w_{a-J}|/|A| = 1$. If the winning category has previously been associated with a different target output, the prediction will be disconfirmed and match-tracking is triggered. So, $\rho_a$ has to be increased to a value slightly greater than unity. This implies that no other nodes can satisfy the $ART_a$ vigilance test and the current input will be ignored.

In view of the above scenario, we propose that during match-tracking, the $ART_a$ vigilance parameter is constrained by

$$0 \le \rho_a \le \min\left(1,\ |A \wedge w_{a-J}|/|A|\right)$$

where $A$ is the current input vector to $F_{2a}$ in complement-coded format and $w_{a-J}$ is the winning $J$th node in $F_{2a}$. Thus, if no other $F_{2a}$ node is able to meet the vigilance test, a new node can be recruited to code the input vector. Now it seems that it is possible to have two similar category nodes to map to different target outputs. This modification is applicable to both the binary and fuzzy ARTMAP networks. Indeed, in the simulations described later which involve only binary data, the fuzzy and binary realisations of ARTMAP are identical.

In FAM, if a tie occurs in the choice function, the winning $F_{2a}$ node is selected in the sequence of 1,2,...[5]. To ensure that similar category nodes have a fair chance to be selected as the winner, we introduce a frequency measure scheme. This frequency measure records the number of correct predictions an $F_{2a}$ category node has accomplished. This information not only facilitates the selection of the winner but also reflects the prior probabilities represented by each $F_{2a}$ category prototype. There are two variants of the frequency measure scheme [10]: INC and INC/DEC based on the reward and reward-penalty rationale. The INC method INCreases the frequency count of each $F_{2a}$ node for correct predictions and no penalty is imposed for incorrect predictions. Conversely, the INC/DEC method INCreases or DECreases the frequency count of each $F_{2a}$ node for correct or incorrect predictions accordingly.

# 3 Simulation Studies

We assess the performance of FAM and modified FAM in classifying *noisy* data set into two classes in stationary and non-stationary environments. The class distributions are fully governed by the prior class probabilities ($P(c_1)$ and $P(c_2)$) and the conditional probabilities or likelihoods ($P(x|c_1)$ and $P(x|c_2)$). By applying Bayes' theorem to these parameters, a data set with a specific posterior probability distribution can be generated. Appendix A shows the parameters used in the following simulations. All the experiments below employ the single-epoch, on-line learning strategy with fast-learning [5,6] and the INC method is adopted for the frequency count. The on-line operational cycle proceeds as follows: an input vector is first presented to $ART_a$ and a prediction is sent to $ART_b$. The predicted output is compared with the actual output and the outcome gives a classification result (prediction). Learning then ensues to cluster the input and target vectors (learning).

### 3.1 Stationary On-line Learning and Classification

Two classes of noisy data samples for the two-bit parity (commonly known as XOR) and four-bit parity problems were generated. In each case, 5000 samples were used and a 1000-sample window was applied to calculate the on-line classification results, *e.g.*, the accuracy at sample 2000 was the number of correct predictions from samples 1001-2000 expressed as a percentage. Although the data statistics are time-invariant, tackling the task on-line is in fact a non-stationary process owing to the build-up of templates—the so-called finite-operating-time problem.

Table 1 shows the average accuracy (Acc.) of 5 runs at the end of the experiment. Although the standard deviation (Std. Dev.) is estimated from a small sample size (5 runs), it does serve as an indication of how the results disperse across the averages. Fig. 2 depicts some typical accuracy plots against increasing number of samples. From the results, it is clear that modified FAM outperforms FAM in all cases with closer proximity to the Bayes limits and smaller standard deviations. Note that some of the results exceed the theoretical Bayes limits because of the use of the window method in calculating the accuracy.

| Bayes Limits | XOR | | | | Four-bit Parity | | | |
|---|---|---|---|---|---|---|---|---|
| | FAM | | Mod. FAM | | FAM | | Mod. FAM | |
| Acc. (%) | Acc. (%) | Std. Dev. | Acc. (%) | Std. Dev. | Acc. (%) | Std. Dev. | Acc. (%) | Std. Dev. |
| 55 | 50.1 | 1.4 | 55.6 | 1.3 | 49.8 | 2.3 | 53.6 | 0.3 |
| 65 | 51.2 | 8.3 | 64.6 | 1.0 | 56.8 | 8.2 | 64.9 | 0.3 |
| 75 | 57.3 | 11.2 | 75.7 | 1.0 | 67.4 | 5.2 | 75.2 | 0.9 |
| 85 | 70.5 | 15.7 | 85.2 | 1.8 | 77.1 | 3.6 | 85.4 | 0.6 |
| 95 | 85.9 | 12.3 | 95.0 | 0.3 | 93.1 | 1.8 | 95.0 | 0.8 |

Table 1 Results for the XOR and four-bit parity problem

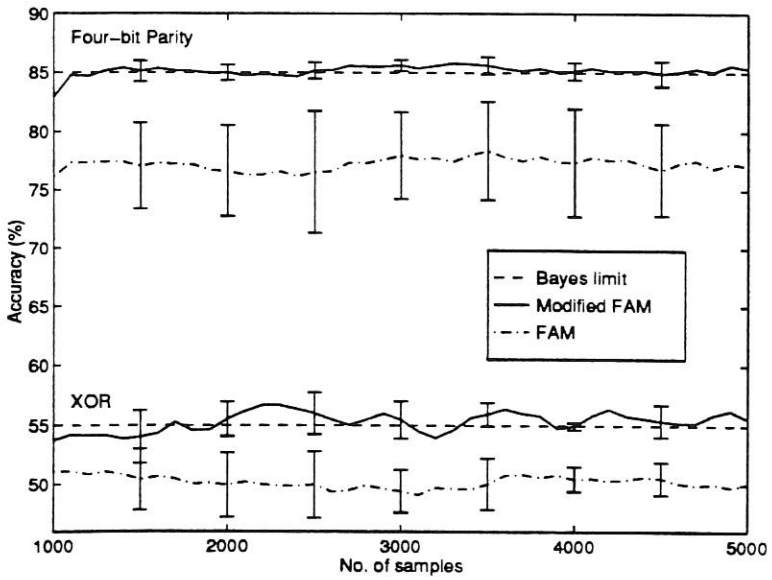


Fig. 2 Modified FAM is able to achieve the Bayes limits more closely than FAM with smaller standard deviations in a stationary environment.

## 3.2 Non-stationary On-line Learning and Classification

In this simulation, two classes of 25000 data samples were generated for the four-bit parity problem. The distribution parameters were subject to step changes every 5000 samples. This simulates a severe non-stationary scenario since in many applications, one might expect to experience a gradual change in the environment rather than a step change. As can be seen in Fig. 3, modified FAM is able to approach the Bayes limit more closely than FAM.

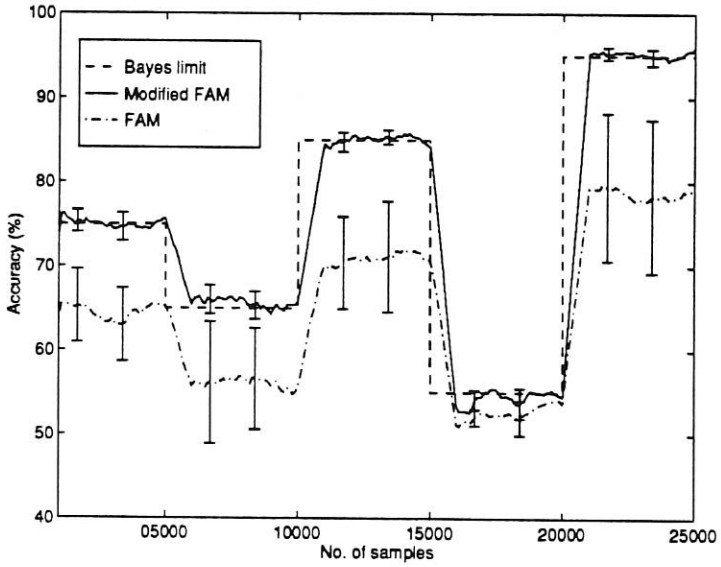


Fig. 3 Modified FAM tracks non-stationarity in the data environment and simultaneously achieves the Bayes limit. The average result of modified FAM also shows smaller standard deviations than FAM.

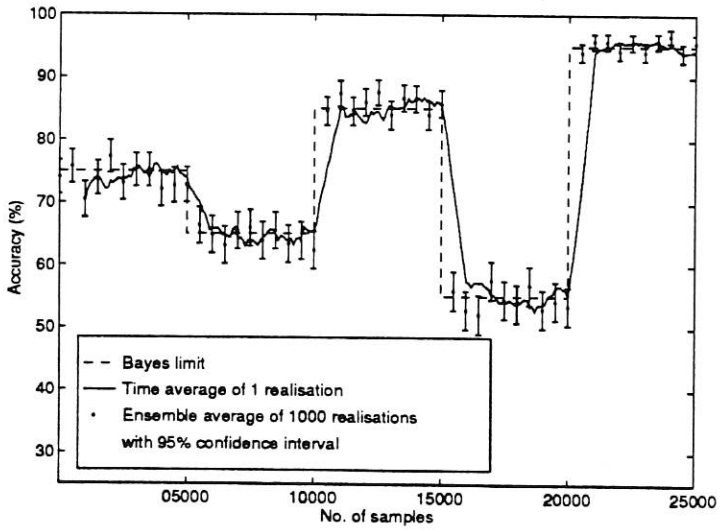


Fig. 4 A comparison between the ensemble and time averages. The Bayes limit and the time average of single realisation fall mostly within the 95% confidence interval of the ensemble average.

### 3.3 Ensemble Average and Time Average

So far we have only used the time average to calculate the on-line results. However, in a stochastic process, each realisation represents only one of the many possible outcomes. Thus, in order to measure the general performance of modified FAM in a non-stationary environment, an ensemble of 1000 networks has been used. The results were calculated across 1000 realisations and compared with the time average of single realisation with a 1000-sample window as in Fig. 4.

## 4 Summary

From the above simulations, modified FAM is able to achieve classification results which closely approximate the Bayes limits in both stationary and non-stationary environments. As might be expected, FAM only creates 4 nodes for the noisy XOR data set and 16 nodes for the four-bit parity data set whereas modified FAM creates 8 and 32 nodes respectively. The formation of category prototypes depends merely on the orderings of input presentation. Once an association has been established for a particular input pattern, FAM will ignore the same pattern when it appears to be a member of a different class. If spurious prototypes (owing to noise) have been developed at the early stage, most of the patterns will be incorrectly classified which directly leads to a degradation in performance. However, owing to the proposed modification, modified FAM is able to create two category nodes in $ART_a$ to map to different target outputs for a specific input pattern. Two identical category nodes are set up with one set serving as the frequently excited prototypes while the other set acts as the spurious prototypes. The frequency measure scheme then ensures that the most probable prototypes are selected to predict an output and thus minimises the overall misclassification rates.

In the non-stationary experiment, modified FAM is competent to recover from drastic changes in data statistics. It tracks the non-stationarity very well and at the same time achieves the Bayes limit. The suitability of using the windowed time average in calculating the results has also been validated with an ensemble of networks. From the comparison, it implies that the time average adequately indicates the ensemble result except close to severe changes in statistics. Note that the time average of single realisation consistently approaches the ensemble average and falls mostly within the 95% confidence interval.

In conclusion, we demonstrate that modified FAM is capable of classifying binary-valued patterns optimally in stationary and non-stationary environments. It creates more category prototypes than the original network in order to implement a one-to-many mapping. Based on the frequency count information, the most probable prototypes are selected to make a prediction which in turn tends to minimise the misclassification rates and thus asymptotically approaches the Bayes limits.

## References


[1] Cybenko, G.: Approximation by Superposition of a Sigmoidal Function. *Mathematics of Control, Signals and Systems*, **2**, pp 303-314. (1989).

[2] Girosi, F., Poggio, T.: Networks and Best Approximation Property. *Biological Cybernetics*, **63**, pp 169-176. (1990).

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

[4] Carpenter, G.A., Grossberg, S.: The ART of Adaptive Pattern Recognition by a Self-Organizing Neural Network. *IEEE Computer*, **21**, pp 77-88. (1988).

[5] Carpenter, G.A., Grossberg, S., Markuzon, N., Reynolds, J.H., Rosen, D.B.: Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimensional Maps. *IEEE Trans. on Neural Networks*, **3**(5), pp 698-712. (1992).

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

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

[8] Moore, B.: ART1 and Pattern Clustering. *Proc. 1988 Connectionist Models Summer School*, pp 174-185. (1988).

[9] Zadeh, L.: Fuzzy Sets. *Information and Control*, **8**, pp 338-353. (1965).

[10] Lim, C.P., Harrison, R.F.: Modified Fuzzy ARTMAP Approaches Bayes Optimal Classification Rates: An Empirical Demonstration. To appear in *Neural Networks*.


## Appendix A

| Bayes Limits | $x$ = Even-parity Samples | | | $x$ = Odd-parity Samples | | |
|---|---|---|---|---|---|---|
| | $P(c_1)$ | $P(x\|c_1)$ | $P(x\|c_2)$ | $P(c_2)$ | $P(x\|c_2)$ | $P(x\|c_1)$ |
| 55% | 0.5 | 0.275 | 0.225 | 0.5 | 0.275 | 0.225 |
| 65% | 0.5 | 0.325 | 0.175 | 0.5 | 0.325 | 0.175 |
| 75% | 0.5 | 0.375 | 0.125 | 0.5 | 0.375 | 0.125 |
| 85% | 0.5 | 0.425 | 0.075 | 0.5 | 0.425 | 0.075 |
| 95% | 0.5 | 0.475 | 0.025 | 0.5 | 0.475 | 0.025 |

Table A1 Parameters used for the XOR data set

| Bayes Limits | $x$ = Even-parity Samples | | | $x$ = Odd-parity Samples | | |
|---|---|---|---|---|---|---|
| | $P(c_1)$ | $P(x\|c_1)$ | $P(x\|c_2)$ | $P(c_2)$ | $P(x\|c_2)$ | $P(x\|c_1)$ |
| 55% | 0.5 | 0.06875 | 0.05625 | 0.5 | 0.06875 | 0.05625 |
| 65% | 0.6 | 0.08125 | 0.04375 | 0.4 | 0.08125 | 0.04375 |
| 75% | 0.4 | 0.09375 | 0.03125 | 0.6 | 0.09375 | 0.03125 |
| 85% | 0.3 | 0.10625 | 0.01875 | 0.7 | 0.10625 | 0.01875 |
| 95% | 0.1 | 0.11875 | 0.00625 | 0.9 | 0.11875 | 0.00625 |

Table A2 Parameters used for the four-bit parity data set