Modified Fuzzy ARTMAP Approaches Bayes Optimal Classification Rates : An Empirical Demonstration

C. P. Lim and R. F. Harrison

Department of Automatic Control and Systems Engineering
University of Sheffield
P.O. Box 600, Mappin Street
Sheffield S1 4DU, UK

Research Report No 515

May 1994

Modified Fuzzy ARTMAP Approaches Bayes Optimal Classification Rates : An Empirical Demonstration

C. P. Lim and R. F. Harrison

Department of Automatic Control and Systems Engineering
University of Sheffield
P. O. Box 600, Mappin Street
Sheffield S1 4DU, UK

Abstract — *This report examines the feasibility of the fuzzy ARTMAP neural network for classifying statistical data and analyses the results according to the Bayes decision criterion. A binary decision with single observation classification problem is chosen to demonstrate and assess the performance of fuzzy ARTMAP. In this task, fuzzy ARTMAP is used to categorise two classes of Gaussian-distributed continuous-valued random variables autonomously and on-line. Various configurations of the task have been investigated by varying the source (mean) separation, prior probabilities and variances of the two Gaussian sources. The results illustrate the limitations of fuzzy ARTMAP in this context. This in turn leads to a modification to the algorithm of fuzzy ARTMAP. Together with a proposed category selection scheme, fuzzy ARTMAP is better able to approach the Bayes optimal classification rates for a binary decision domain.*

1 Introduction

Pattern classification is an active research area in neural networks. Cybenko (1989) argued that network architectures using logistic functions are able to approximate any function with arbitrary accuracy. A similar finding is also concluded for radial basis function networks, e.g. Poggio & Girosi (1990) and Light (1992) showed that a radial basis function network can approximate any multivariate continuous functions with a sufficient number of radial basis function units. This is an essential characteristic for establishing nonlinear decision boundary surfaces by neural networks in pattern classification. Recently, many neural networks have been used as classifiers and their outputs have been interpreted as estimates of the Bayesian posterior probabilities (White 1989, Wan 1990). Some of the neural networks are developed based on the Bayes strategy for pattern classification (Specht 1990, Hrycej 1992, Musavi *et al* 1993). Simulation results from multilayer perceptron network trained with back-propagation, radial basis function network and higher order polynomial network have shown that the network outputs provide good estimates of the Bayesian posterior probabilities (Richard & Lippmann 1991).

When developing a neural network classifier, we typically proceed by gathering a set of data to train the network. Information is encoded during the training cycle by the adjustment of weights. After that, the network is put into operation and no further adaptation (learning) is permitted. Moreover, when the network is presented with a previously unseen input pattern, there is generally no built-in mechanism for the network to recognise the novelty. Thus, if we wish to add new information to the

network, we would have to re-train the network using the new data together with all previous data. This is the major drawback suffered by most neural network architectures and is expressed by the stability-plasticity dilemma (Carpenter & Grossberg 1988). It poses the questions: how can a learning system remain plastic or adaptive in response to significant events and yet remain stable in response to irrelevant events? How can a system adapt to new information without corrupting or forgetting previously learned information? In response to this dilemma, Carpenter, Grossberg and colleagues have developed a family of neural network architectures called Adaptive Resonance Theory (ART).

ART is implemented in various versions : ART1, ART2, ART3 and fuzzy ART for unsupervised learning; ARTMAP and fuzzy ARTMAP for supervised learning (Carpenter *et al* 1987a, 1987b, 1990, 1991a, 1991b, 1992). ARTMAP has been reported to be able to classify binary and analog patterns in real-time and nonstationary environments (Carpenter *et al* 1991b, 1992). However, its performance is yet to be examined in classifying statistical data and to interpret the results according to probability theory. This work investigates the feasibility of fuzzy ARTMAP in separating two classes of Gaussian distributed random variables and compares the results with the Bayes decision criterion. The limitations of fuzzy ARTMAP in this context are explained and a novel modification is proposed which enables fuzzy ARTMAP to better approach the Bayes optimal classification rates autonomously and on-line.

Section 2 reviews some criteria for designing binary decision rules. Section 3 presents a summarised description of ART and, in particular, fuzzy ARTMAP together with the modifications to its algorithm. Section 4 reports and discusses the simulation results and a conclusion follows.

2 Binary Decision

Binary decision with single observation is the simplest case in any decision-making process. Despite its simplicity, binary decision problems illustrate most of the fundamental concepts underlying all statistical decision theories. By binary decision, we mean that there are two classes of message, c_1 and c_2, corresponding to two decisions, d_1 and d_2, i.e., if c_1 is selected, then d_1 is the decision; and if c_2 is selected, then d_2 is the decision. The problem is that given an observation or input x from the input space, a decision rule d(x) has to be determined such that x is mapped into the decision space in some optimal manners (Melsa & Cohn, 1978). The mapping criterion is usually related to the minimisation of misclassification. Figure 1 illustrates a schematic diagram of a binary decision classification problem. Since there are only two decisions, this is equivalent to dividing the input space into two decision regions, X_1 and X_2, such that $d(x) = d_1$ if $x \in X_1$ and $d(x) = d_2$ if $x \in X_2$.

The task of separating two Gaussian sources can be viewed as a binary decision problem. The two messages correspond to the two classes, c_1 and c_2, of Gaussian distributed random variables with, in general, differing means, variances and prior probabilities. The observation or input corresponds to a continuous-valued random variable, x, which may belong to one of the two classes. The objective is, therefore, to choose the decision boundary which minimises the probability of misclassification.

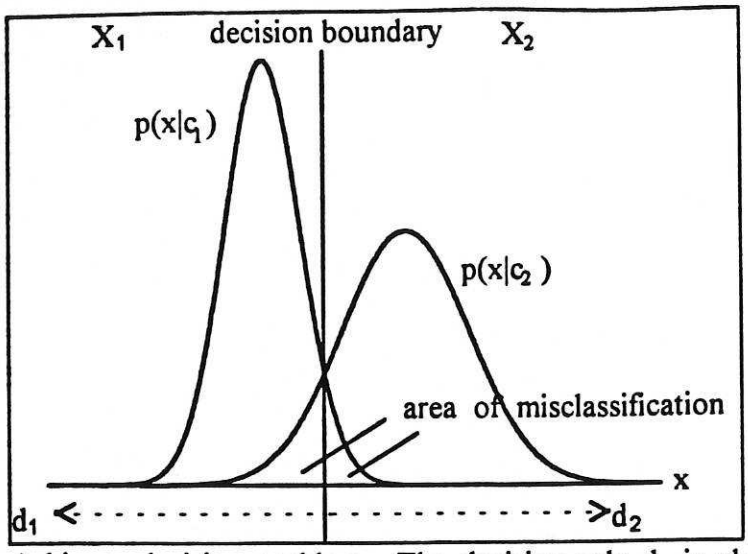

Figure 1 A binary decision problem. The decision rule d_1 is chosen when the conditional probability density $p(x|c_1) > p(x|c_2)$ and vice versa. This is equivalent to dividing the input space into X_1 and X_2.

2.1 Maximum Likelihood Decision Criterion

Before illustrating the formalism of the Bayes decision criterion, first examine the simplest technique for designing a binary decision rule, i.e., the maximum likelihood criterion. Consider the two-class problem where an input x is known to be either in c_1 or c_2. The maximum likelihood criterion states that we should decide $d(x) = d_1$ if it is more likely that c_1 generated x than c_2 generated x,

$$d(x) = \begin{cases} d_1 & \text{if} \quad p(x|c_1) > p(x|c_2) \\ d_2 & \text{if} \quad p(x|c_2) > p(x|c_1) \end{cases}$$

where $p(x|c_i)$ is the conditional probability density of x given c_i, i = 1, 2.

As depicted in figure 1, any binary decision problem is equivalent to dividing the input space into two decision regions, X_1 and X_2, i.e.

$$X_1 = \{x: p(x|c_1) > p(x|c_2)\} \tag{1}$$
$$X_2 = \{x: p(x|c_2) > p(x|c_1)\} \tag{2}$$

Define the likelihood ratio $\Lambda(x)$ as

$$\Lambda(x) = \frac{p(x|c_2)}{p(x|c_1)}$$

then equations (1) and (2) become

$$X_1 = \{x: \Lambda(x) < 1\}$$
$$X_2 = \{x: \Lambda(x) > 1\}$$

$$\text{or} \quad \Lambda(x) \underset{d_1}{\overset{d_2}{\gtrless}} 1 \tag{3}$$

Note that from equation (3) the maximum likelihood criterion makes a decision by comparing the likelihood ratio with unity. This simplicity often seems inadequate to represent some realistic problems. For instance, in a medical diagnosis situation, several symptoms have been observed to determine if a patient has disease A or B. Assume that the symptoms have probability x associated with disease A and probability y associated with disease B. Based on the maximum likelihood criterion, if $x > y$, the patient would be diagnosed to have disease A. However, if disease A is rare and disease B is common, then the decision does not reflect the real statistical information about the disease. In other words, the prior probability of an event plays a role in determining the decision rule.

2.2 Bayes Decision Criterion

A traditional goal for decision strategies used in classification is that they minimise the "expected risk" (Dunteman 1984, Fu 1982). The Bayes decision criterion employs a systematic procedure of assigning a cost to each correct and incorrect decision and then minimising the total average or expected cost. Let C_{jk} be the cost of making decision d_j when c_k is correct. For a binary decision problem, the expected cost is

$$\begin{aligned} E\{C_{jk}\} = {} & C_{11}P\{d_1|c_1\}P\{c_1\} + C_{12}P\{d_1|c_2\}P\{c_2\} + \\ & C_{21}P\{d_2|c_1\}P\{c_1\} + C_{22}P\{d_2|c_2\}P\{c_2\} \end{aligned} \tag{4}$$

where $P\{c_i\}$ is the prior probability of class c_i, i = 1,2 and $P\{d_i|c_j\}$ is the probability of making decision d_i when c_j is true, i=1,2, j=1,2. Note that

$$P\{d_i|c_j\} = P\{x \in X_i|c_j\} = \int_{X_i} p(x|c_j)dx \tag{5}$$

$$P\{d_1|c_1\} = 1 - P\{d_2|c_1\} \tag{6}$$

$$P\{d_1|c_2\} = 1 - P\{d_2|c_2\} \tag{7}$$

From equations (5), (6) and (7), equation (4) becomes

$$\begin{aligned} E\{C_{jk}\} = {} & C_{11}P\{c_1\} + (C_{21} - C_{11})P\{d_2|c_1\}P\{c_1\} + \\ & C_{12}P\{c_2\} - (C_{12} - C_{22})P\{d_2|c_2\}P\{c_2\} \end{aligned}$$

$$\begin{aligned} E\{C_{jk}\} = {} & C_{11}P\{c_1\} + C_{12}P\{c_2\} + (C_{21} - C_{11})P\{c_1\}\int_{X_2} p(x|c_1)dx - \\ & (C_{12} - C_{22})P\{c_2\}\int_{X_2} p(x|c_2)dx \end{aligned}$$

$$\begin{aligned} E\{C_{jk}\} = {} & C_{11}P\{c_1\} + C_{12}P\{c_2\} + \int_{X_2} [(C_{21} - C_{11})P\{c_1\}p(x|c_1) - \\ & (C_{12} - C_{22})P\{c_2\}p(x|c_2)]dx \end{aligned} \tag{8}$$

The Bayes decision criterion states that we should select the decision region X_2 such that the expected cost $E\{C_{jk}\}$ given by equation (8) is minimised. Since the first two terms in equation (8) are not a function of X_2, to minimise the expected cost the integral must be

$$(C_{21}-C_{11})P\{c_1\}p(x|c_1)-(C_{12}-C_{22})P\{c_2\}p(x|c_2)<0$$

and the decision rule would be

$$(C_{12}-C_{22})P\{c_2\}p(x|c_2) \underset{d_1}{\overset{d_2}{\gtrless}} (C_{21}-C_{11})P\{c_1\}p(x|c_1)$$

or

$$\Lambda(x) \underset{d_1}{\overset{d_2}{\gtrless}} \frac{(C_{21}-C_{11})P\{c_1\}}{(C_{12}-C_{22})P\{c_2\}} \qquad (9)$$

By inspecting equations (3) and (9), instead of comparing the likelihood ratio with unity as in the maximum likelihood decision, the Bayes decision criterion takes into account the ratio of prior probability of each class associated with their respective costs in the likelihood ratio test.

2.3 Error and Accuracy

In this work, we decided to assign unit cost for misclassification and no cost for correct decision, i.e.,

$$C_{11}=C_{22}=0$$
$$C_{12}=C_{21}=1$$

The likelihood ratio test in equation (9) reduces to the ratio of prior probability of the two classes.

$$\Lambda(x) \underset{d_1}{\overset{d_2}{\gtrless}} \frac{P\{c_1\}}{P\{c_2\}}$$

The misclassification rate or the probability-of-error is calculated as

$$P_e = P\{d_2|c_1\}P\{c_1\}+P\{d_1|c_2\}P\{c_2\}$$
$$P_e = P\{c_1\}\int_{X_2} p(x|c_1)dx + P\{c_2\}\int_{X_1} p(x|c_2)dx$$

and the classification accuracy is defined as

$$\text{Accuracy} = 1-P_e$$
$$\text{Percentage of Accuracy} = (1-P_e)x100\%$$

3 Adaptive Resonance Theory

This section starts with a typical pattern-matching scenario in ART and a summarised description on fuzzy ART and fuzzy ARTMAP. It then explains the problem of one-to-many mapping in ARTMAP. The problem was revealed when attempting to establish one-to-many classification during the experiment. This in turn led to a modification to the fuzzy ARTMAP algorithm and the motivation of incorporating a frequency measure scheme to the prototypes in an ART system.

3.1 Pattern-matching in ART

ART is a self-organising neural network architecture capable of categorising input patterns into different recognition categories (Carpenter *et al*, 1987a). Figure 2 illustrates the main components of an ART module. It consists of two parts: the attentional subsystem and the orienting subsystem. The attentional subsystem has two layers, F_1 and F_2, of processing neurons called nodes and a gain control to each layer. The orienting subsystem has a reset circuit which plays an important role in determining the classification results. Notice that there are three possible input sources to F_1 and F_2. The nodes in F_1 and F_2 obey the 2/3 rule, i.e., they become active only if at least two of the three sources are active (Carpenter *et al*, 1987a).

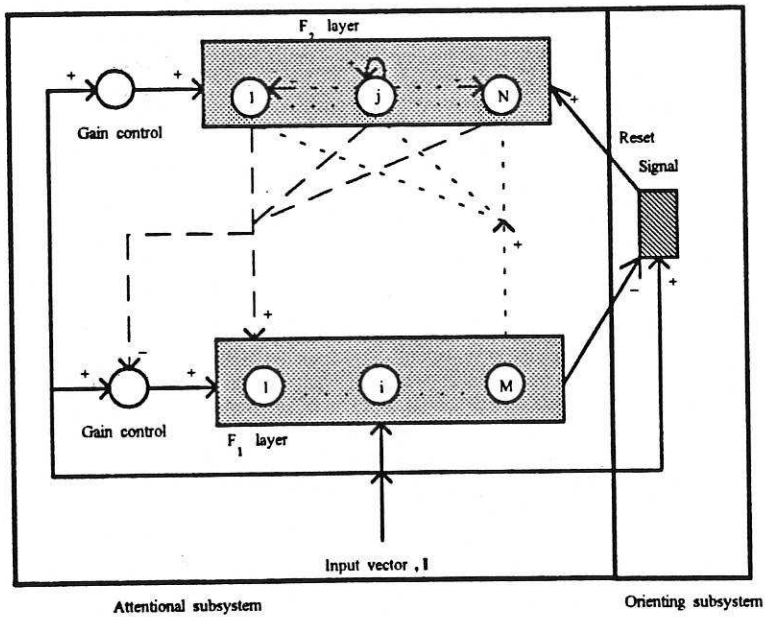

Figure 2 A general schematic diagram for an ART system. Nodes in F1 and F2 are fully interconnected. A (+) indicates an excitatory connection whereas a (-) indicates an inhibitory connection.

Figure 3 presents a typical pattern-matching cycle in ART. An input vector **I** registers itself as a pattern in the F_1 layer. A pattern of activation **X**, i.e., the Short Term Memory (STM) activity, is produced across F_1. The appearance of STM results in an output pattern **U** to be transmitted from F_1 and F_2 via the adaptive bottom-up weights, or Long Term Memory (LTM) traces. Each F_2 node receives the entire vector **U** weighted by the LTM and makes a response to this stimulus. Due to the internal competition dynamics of self-reinforcement and lateral inhibition (on-center off-surround competition), the node in F_2 which responds with the largest activation is

chosen to be the winner (winner-take-all). A STM pattern **Y** is formed across F_2 while other nodes are shut down. The winning node then sends its prototype vector **P** to F_1 via the top-down LTM traces. This in turn leads to a new pattern of STM, **X***, in F_1. The similarity between the STM vector, **X***, and the input vector, **I**, is tested by the reset circuit against the vigilance threshold ρ. If the vigilance test fails to meet the criterion, a signal is sent by the reset circuit to inhibit the winning F_2 node for the rest of the pattern-matching cycle. Now, **I** is reinstated at F_1 and the search continues until an F_2 node satisfies the vigilance test. If no such node exists, a new node is created in F_2 to code the input pattern.

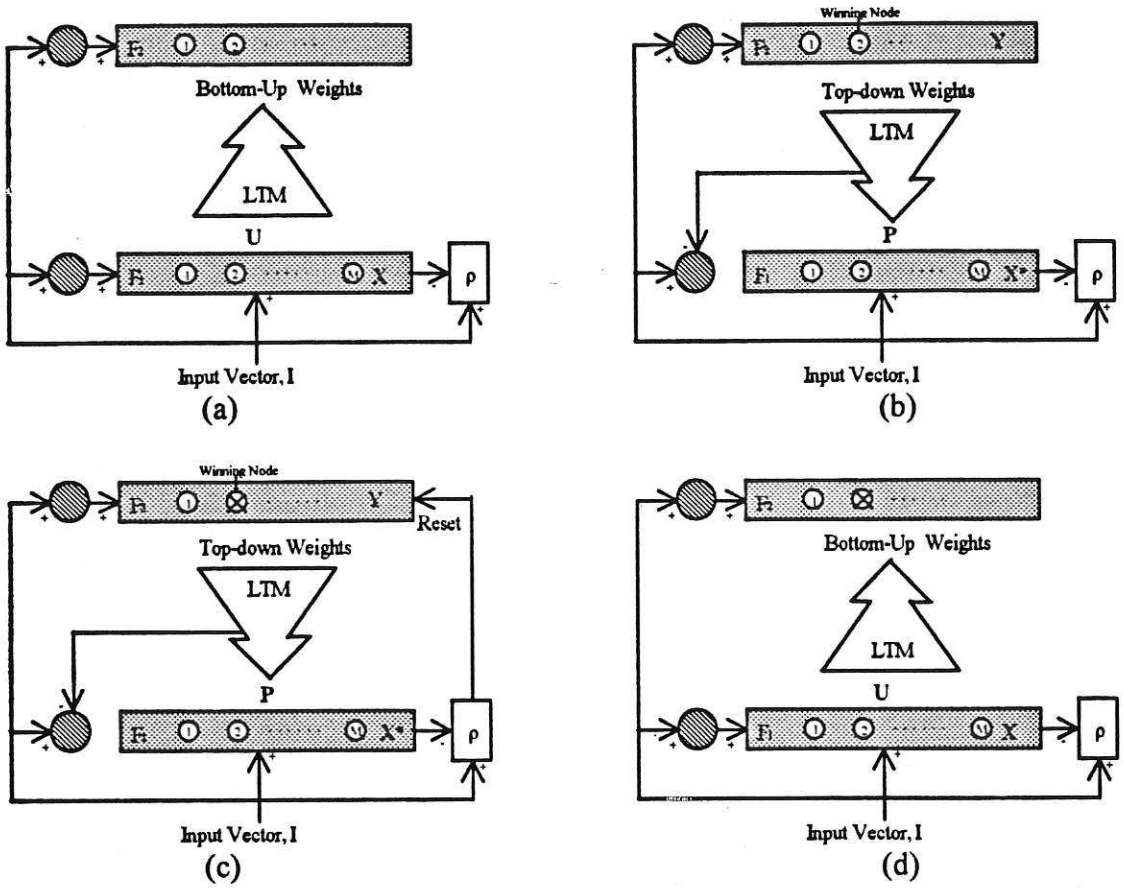

Figure 3 A typical pattern matching scenario in an ART network. (a) An input vector **I** goes to F_1 and induces a STM pattern which results in a stimulus to be transmitted to F_2 via the bottom-up LTM traces. (b) Based on the responses, a winning node in F_2 is selected and a prototype is sent to F_1 via the top-down LTM traces. (c) Mismatch between the prototype and the input initiates a reset signal to deactivate the winning node. (d) The input vector is re-applied to F_1 to start a new search.

3.2 Fuzzy ART

Fuzzy ART (Carpenter *et al* 1991a) is a generalisation of ART1 (Carpenter & Grossberg 1987a) for unsupervised learning. It incorporates fuzzy set theory into ART1 by replacing the intersection operator ($\cap$) in ART1 by the fuzzy MIN operator ($\wedge$). The MIN operator reduces to the intersection operator in binary cases and thus enables fuzzy ART to handle analog, binary and fuzzy input patterns.

In general, the algorithm of fuzzy ART can be divided into the following phases:

(a) Initialisation

Referring to figure 2, a fuzzy ART system has two layers, F_1 and F_2. F_1 receives an input vector $\mathbf{I}=(I_1,...,I_M)$ in the manner that each vector component goes to one node. Each component I_i, for i = 1 to M, should be in the interval [0,1]. F_2 is a layer containing the category prototypes, i.e., each node representing a cluster of input patterns. Note that this is a dynamical field where nodes can be created when necessary, thus allowing the number of category prototypes to grow arbitrarily. Associated with each F_2 node (indexed by j, for j = 1, 2,...) is a vector of weights representing the LTM traces. The weight vector is initialised as

$$w_{j1}(0) = ... = w_{jM}(0) = 1 \tag{10}$$

The weight vectors **w** subsume both the bottom-up and top-down LTM traces of ART1. Initially, each F_2 node is uncommitted. When it learns a category prototype, it becomes committed by modifying its weight vector to code the input pattern.

There are three parameters which control the dynamics of fuzzy ART, i.e., a choice parameter $\alpha > 0$, a learning parameter $\beta \in [0,1]$, and a vigilance parameter $\rho \in [0,1]$. The choice parameter α should take a small value. If α is large, there is a tendency to choose an uncommitted node before going into a deeper search for previously committed F_2 nodes (Carpenter & Grossberg 1987a). Hence, $\alpha \rightarrow 0$ is known as the conservative limit which tends to minimise recoding of prototypes during learning. The learning parameter affects how learning takes place, e.g. fast learning or fast-commit slow-recode learning. The vigilance parameter determines the coarseness of the classification, i.e., how "similar" do we want a cluster of input patterns to be?

(b) Propagation

The input vector activates a STM pattern, **X**, across F_1 and an output vector **U** is sent to F_2. The vector **U** is the same as the input vector,

$$\mathbf{U} = \mathbf{I}$$

(c) Recognition

Each F_2 node responds differently to the input vector according to a choice function

$$T_j(\mathbf{I}) = \frac{|\mathbf{I} \wedge \mathbf{w}_j|}{\alpha + |\mathbf{w}_j|}$$

The fuzzy MIN operator and the norm |.| is defined as

$$(\mathbf{a} \wedge \mathbf{b})_i = \min(a_i, b_i)$$

$$|\mathbf{a}| = \sum_{i=1}^{M} |a_i|$$

The choice function measures the match between the input vector and the weight vector according to the fuzzy subsethood theory (Zadeh 1965). The node with the highest response, denoted as the Jth node, is selected as the winner while all other nodes are shut down. A STM pattern in F_2 is formed, i.e.,

$$y_j = \begin{cases} 1 & \text{if} \quad T_J = \max\{T_j : j = 1,2,\ldots\} \\ 0 & \text{otherwise} \end{cases}$$

If there is a tie in T_j, the smallest index j is chosen. The winning node is now required to send its stored prototype back to F_1.

(d) Feedback

The prototype vector **P** is actually the weight vector of the winning F_2 node, $\mathbf{w}_J$.

$$\mathbf{P} = \mathbf{w}_J$$

It induces a new STM activity, **X***, across F_1 as defined by

$$\mathbf{X}^* = \mathbf{I} \wedge \mathbf{w}_J$$

(e) Comparison

Both the F_1 STM vector, **X***, and the input vector, **I**, are passed to the reset circuit to test their resemblance. This vigilance test is governed by a match function,

$$\frac{|\mathbf{X}^*|}{|\mathbf{I}|} = \frac{|\mathbf{I} \wedge \mathbf{w}_J|}{|\mathbf{I}|} \geq \rho$$

Resonance is said to occur if the vigilance criterion is met, and learning ensues.

(f) Search

However, if the vigilance test fails, i.e.,

$$\frac{|\mathbf{X}^*|}{|\mathbf{I}|} = \frac{|\mathbf{I} \wedge \mathbf{w}_J|}{|\mathbf{I}|} < \rho$$

a reset signal is sent to F_2. The winning F_2 node is disabled and the choice function is set to zero. This has a twofold effect: the node is prevented from entering any further best-match competitions for the current input pattern, and the F_2 layer is reset to allow new activity. The input vector is now re-applied to F_1 and the pattern-matching cycle continues. This process is repeated, consecutively disabling nodes in F_2, until a category is found that satisfies the vigilance test. If no such node exists, a new node is created in F_2 to code the input vector.

(g) Learning

Once search ends, learning takes place by adjusting the weight vector $\mathbf{w}_J$ according to

$$\mathbf{w}_J^{(new)} = \beta(\mathbf{I} \wedge \mathbf{w}_J^{(old)}) + (1-\beta)\mathbf{w}_J^{(old)}$$

Fast learning corresponds to setting $\beta = 1$. However, for efficient coding of noisy input sets, it is useful to employ the fast-commit slow-recode learning rule (Carpenter *et al*, 1991a). Then, learning is accomplished by setting $\beta = 1$ when J is an uncommitted node and $\beta < 1$ after the node is committed.

3.2.1 Complement Coding

Moore (1989) reported that ART1 could be subjected to a category proliferation problem. As a result, Carpenter *et al* (1991a) proposed a normalisation method to the input vectors called complement coding so that the category proliferation problem is avoided. Complement coding is a normalisation rule that preserves amplitude information. It represents both the on-response and the off-response to an input vector. To represent such a code in its simplest form, let an incoming vector **a** represents the on-response and the complement of **a**, denoted as $\mathbf{a}^c$, represents the off-response, where

$$a_i^c = 1 - a_i$$

Thus, the complement coded input **I** to the F_1 field is a 2M-dimensional vector,

$$\mathbf{I} = (\mathbf{a}, \mathbf{a}^c) = (a_1, \ldots, a_M, a_1^c, \ldots, a_M^c)$$

with each component I_i in the interval [0, 1] and the norm of **I**

$$|\mathbf{I}| = |(\mathbf{a}, \mathbf{a}^c)| = \sum_{i=1}^{M} a_i + (M - \sum_{i=1}^{M} a_i) = M$$

When complement coding is used, the initial condition of equation (10) is replaced by

$$w_{j1}(0) = \ldots = w_{j,2M}(0) = 1$$

3.3 Fuzzy ARTMAP

While fuzzy ART is an unsupervised learning system, fuzzy ARTMAP (Carpenter *et al*, 1992) is a supervised learning system capable of classifying binary, analog and fuzzy input patterns into recognition categories autonomously based on predictive success. Fuzzy ARTMAP consists of a pair of fuzzy ART modules, ART_a and ART_b, that are linked by an inter-ART module called the map field. Figure 4 depicts the main components of a fuzzy ARTMAP system.

During operation, ART_a reads an input pattern **a** and ART_b reads another pattern **b** where **b** is the correct/target output of **a**. Each fuzzy ART module self-organises in response to their input vectors. A map field controls the learning of an associative map from ART_a recognition categories (ART_a F_2 layer) to ART_b recognition categories (ART_b F_2 layer). However, this map field does not directly associate input **a** with **b**, but rather links the compressed and symbolic representations of the prototypes of **a** and **b** (Carpenter *et al*, 1991b) We now concentrate on how the two modules of fuzzy ART are connected by the map field to operate as a supervised learning system.

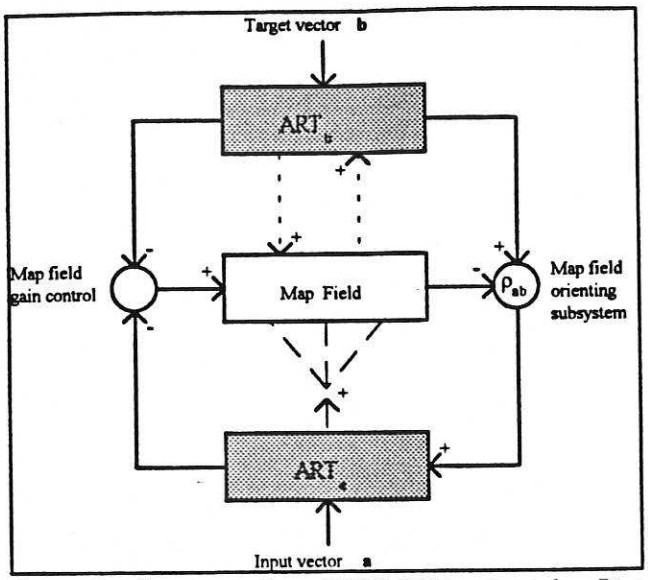

Figure 4 A schematic diagram of an ARTMAP network. It consists of a pair of ART modules interconnected via a map field.

3.3.1 The Map Field

Figure 5 illustrates the main components of the map field. F_{2a} is the F_2 layer of ART_a, F_{2b} is the F_2 layer of ART_b and F_{ab} is the map field layer. Let j , j =1, 2, ..., be the index for the nodes in F_{2a} and k, k = 1, 2,..., be the index for the nodes in F_{2b}. The number of nodes in F_{ab} is the same as the number of nodes in F_{2b}. Note that there is a one-to-one permanent link between the nodes in F_{2b} and F_{ab}. But, F_{2a} is linked to F_{ab} via an adaptive pathway of map field weight vectors.

(a) Map field initialisation

The map field weight vectors are initialised to

$$w_{jk}(0) = 1$$

These weight values imply that every node in F_{2a} can be linked to every node in F_{2b} via F_{ab}, i.e., no predictive association exists initially.

(b) Map field association

ART_a and ART_b categorise their input vectors separately and independently. After resonance occurs in each module, there is an active category prototype (winning node) in F_{2a} and F_{2b} (denoted as node J and node K). F_{2a} and F_{2b} both send their signals, $\mathbf{y}_a$ and $\mathbf{y}_b$, to the map field. An association is formed by setting the map field STM $\mathbf{x}_{ab}$ to

$$\mathbf{x}_{ab} = \mathbf{y}_b \wedge \mathbf{w}_{ab-J} \tag{11}$$

when both F_{2a} and F_{2b} are active. Note that $\mathbf{w}_{ab-J}$ is the map field weight vector associated with the Jth winning node in F_{2a}. If the prediction ($\mathbf{w}_{ab-J}$) is disconfirmed by F_{2b} ($\mathbf{y}_b$), then $\mathbf{x}_{ab} = 0$. This triggers a match tracking activity.

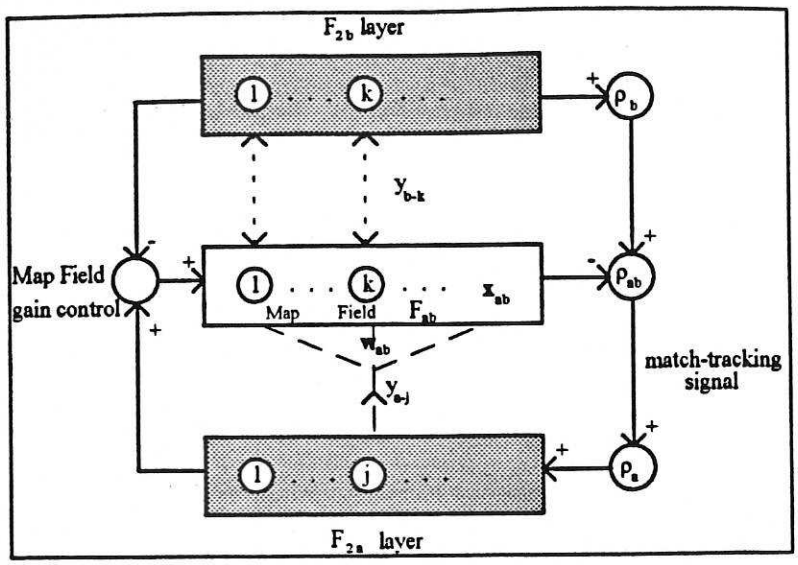

Figure 5 A permanent one-to-one connection exists from ART_b F_2 (F_{2b}) to the map field (F_{ab}); whereas ART_a F_2 (F_{2a}) is connected to F_{ab} via an adaptive pathway of map field LTM traces (w_{ab}). When both ART_a and ART_b are active, the map field receives a predicted signal y_{a-j} from the jth node in F_{2a} and an answer y_{b-k} from the kth node in F_{2a}. If the prediction is disconfirmed, a match-tracking signal is sent to ART_a to start a search.

(c) Map field reset and match-tracking

The mismatch event resets the map field and activates the control strategy called match-tracking. Match-tracking regulates ρ_a (ART_a vigilance parameter) in such a way as to keep the system from making repeated errors. Initially, for a new input pattern ρ_a is relaxed to a baseline vigilance. When a predictive error occurs, i.e.,

$$|\mathbf{x}_{ab}| < \rho_{ab}|\mathbf{y}_b|$$

where ρ_{ab} is the map field vigilance parameter, ρ_a is increased to a value just enough to trigger a search in the ART_a module. Thus ρ_a should be raised to a value slightly greater than $|\mathbf{a} \wedge \mathbf{w}_{a-J}|/|\mathbf{a}|$ to cause the ART_a vigilance test to fail, i.e.,

$$|\mathbf{a} \wedge \mathbf{w}_{a-J}| < \rho_a|\mathbf{a}|$$

By increasing ρ_a match-tracking provides a means to select a node in F_{2a} which fulfils both

$$|\mathbf{a} \wedge \mathbf{w}_{a-J}| \geq \rho_a|\mathbf{a}|$$
$$|\mathbf{x}_{ab}| \geq \rho_{ab}|\mathbf{y}_b|$$

If no such node exists, F_{2a} is shut down for the rest of the input presentation and the input pattern is ignored.

(d) Map field learning

When a node J in F_{2a} successfully predicts a node K in F_{2b}, learning is carried out by linking node J to node K according to

$$w_{ab-J} = x_{ab} \tag{12}$$

From equation (11) and (12), $w_{ab-J} = 1$ for all time, indicating that a permanent association is made to allow node J to predict node K in future.

3.4 One-to-Many Mapping

In general, there are two types of neural network-based classifiers (Ryan 1988) :

(1) networks with interconnection weights that can be interpreted as cluster/category prototypes.

(2) networks with weights that interpolate the decision surfaces separating pattern clusters/categories.

ART is an example of the first type whereas networks using, say, back-propagation learning fall into the second type.

As explained in section 3.1, ART employs the "winner-take-all" competition scheme in the F_2 layer. The input pattern is compared with all category prototypes (committed nodes) in F_2 and the one that best matches the input pattern is selected as the winner. Other nodes are shut down. Hence, the predictive results of fuzzy ARTMAP depend heavily on the category prototypes in the F_2 layers of ART_a and ART_b. Note that each category is in fact the prototype of a cluster of patterns defined by the vigilance threshold (Burke 1991, Moore 1989).

In the paper by Carpenter *et al* (1991b), it is stated that one-to-many learning is possible in ART. An example of associating the taste of bananas with different features is given. It is argued that an input pattern can be associated with many learned category prototypes, each representing a feature of the input pattern. This is carried out by using predictive feedback during the hypothesis testing cycle where attention can be shifted to new recognition category without recoding previously learned categories. Eventually, an ART system will select the one that best describes the input pattern to be the representative prototype. Hence, one-to-many recognition and prediction codes can be formed through time.

However, this one-to-many learning scheme is confined to single ART modules, and in ARTMAP it does not mean that a prototype in F_{2a} can be linked to many prototypes in F_{2b} via the map field. The reason for this is to avoid any confusion of mapping, and hence the prediction, from F_{2a} recognition categories to F_{2b} predictive answers. In other words, it is not known which category in F_{2b} should be selected as the output if more than one association exists. This phenomenon is clearly stated in the papers on ARTMAP and fuzzy ARTMAP (Carpenter *et al* 1991b, 1992). During the match-tracking activity (i.e., the current winning ART_a category fails to match the active ART_b category), the vigilance parameter of ART_a is increased to a value that causes the vigilance test to fail. This process leads either to the activation of another ART_a

category which is able to satisfy both ART_a and map field vigilance tests *or* to the shutdown of F_{2a} until the input pattern is removed.

In a pattern classification task, the probability density functions of the data can sometimes be densely overlapped. In this experiment, input to ART_a is a random number x and input to ART_b is the target class of x. ART_a will segregate the input range of x into several sub-ranges and assign one node to code a prototype of each sub-range. The allowable "width" of the sub-ranges is defined by the vigilance parameter. Figure 6 depicts one of the possible sub-ranges denoted by broken lines. For any x that fall within that sub-range, it may belong to either c_1 or c_2. In view of this, it is desirable to establish one prototype that can be mapped to either c_1 or c_2, thus a one-to-many mapping. This problem motivated the modification of the fuzzy ARTMAP algorithm to enable a one-to-many mapping.

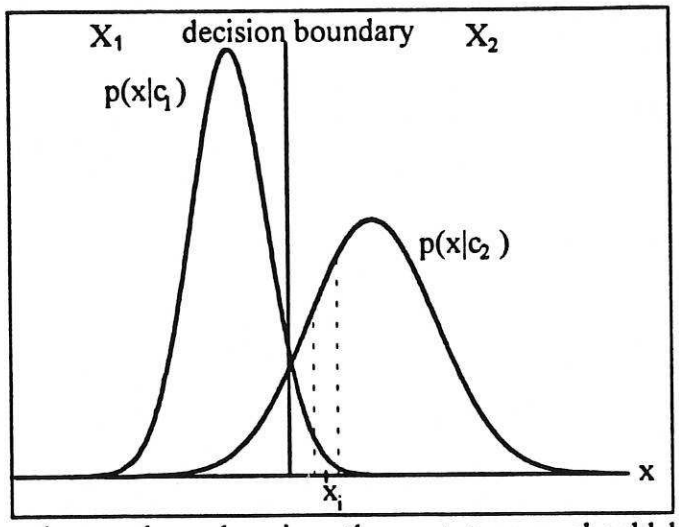

Figure 6 For the overlapped region, the prototype x_i should be able to predict both c_1 and c_2. One-to-many mapping attempts to associate this prototype with different predicted outputs.

3.5 Modified Fuzzy ARTMAP

One way to accomplish a one-to-many mapping is proposed as follows. We shall work through an example to illustrate the method. Referring to figure 6, first assume that the input pair is $\mathbf{x}_i$ and c_1 presented in complement coded format. A prototype of the vector $\mathbf{x}_i$ will be set up in F_{2a} which is linked to predict c_1 in F_{2b} i.e.,

$$\mathbf{w}_J = \mathbf{x}_i \tag{13}$$

where $\mathbf{w}_J$ is the weight vector of the Jth winning node in F_{2a}.

Then, an input pair of $\mathbf{x}_i$ and c_2 arrives. $\mathbf{w}_J$ is selected and the predicted class is c_1. However, this prediction is disconfirmed by ART_b because the actual class is c_2, and hence match-tracking is triggered. Since it is a perfect match of the input and the prototype, the vigilance parameter of ART_a (ρ_a) is increased to a value slightly greater than

$$\frac{|\mathbf{x}_i \wedge \mathbf{w}_J|}{\mathbf{x}_i} = 1 \qquad (14)$$

From equation (13) and (14), we can see that no other nodes, even a new uncommitted node, in F_{2a} can satisfy the vigilance test because ρ_a is greater than unity. So, the input pattern will be ignored.

Instead of increasing the baseline vigilance to a value greater than 1, it is set to 1. This implies that a new uncommitted node will be selected to code the input pattern $\mathbf{x}_i$ and associate it with c_2.

By generalising this idea, during match-tracking the ART_a vigilance parameter is constrained by

$$0 \leq \rho_a \leq \min(1, \frac{|\mathbf{a} \wedge \mathbf{w}_J|}{|\mathbf{a}|}) \qquad (15)$$

where **a** is the current input vector to ART_a and $\mathbf{w}_J$ is the Jth winning node in F_{2a}.

According to equation (15), if no committed node in F_{2a} is able to satisfy the vigilance test, a new node is selected to code the input vector. By using this approach, it is possible to create two "similar" (within the range of vigilance), if not identical, prototypes which map to different recognition categories in F_{2b}, thus implementing a one-to-many mapping.

There are two main disadvantages with this approach.

(1). Redundancy of Category Prototypes
It is obvious that two or more prototypes are produced to map an input cluster to different outputs. This will increase the number of nodes in F_{2a} but many of the prototypes will be redundant.

(2). Selection of Category Prototypes
Assume that there are two identical prototypes in F_{2a}. When an input pattern arrives, which prototype should be selected as the winner? In the present implementation, a prototype in F_{2a} is selected in sequence i.e., 1,2,... . In the above example, let node 5 in F_{2a} be the prototype of $\mathbf{x}_i$ associated with c_1 and node 8 be the prototype of $\mathbf{x}_i$ associated with c_2 . Hence, c_1 will always be the predicted output because 5 precedes 8. Consequently, the misclassification rate is increased because the conditional probability density $p(x|c_2)$ is greater than $p(x|c_1)$ for that particular sub-range.

3.6 A Frequency Measure for ART Prototypes

The problem of prototype selection motivates the introduction of a frequency measure for the learned prototypes to facilitate the selection between "tied" nodes. The approach measures the frequency of each prototype being associated with a predicted answer.

In general, the operation of fuzzy ARTMAP on-line classification can be divided into two phases: the prediction phase and the learning phase. First, an input vector is presented to ART_a. A winning node is selected in F_{2a} which makes a prediction to ART_b via the map field. The actual output is then fed to ART_b. The predicted output is compared with the actual output to determine its correctness. This gives a result for the classification. If the prediction is confirmed, the frequency count of the winning F_{2a} node is incremented by one. However, if the prediction is incorrect, the frequency count is reduced by one.

The same procedure applies in the learning phase. If the predicted category is disconfirmed by ART_b, match-tracking is triggered and a search in ART_a is initiated. When a category prototype in F_{2a} correctly predicts an output in ART_b, its frequency count is incremented by one. Similarly, the frequency count of the winning F_{2a} prototype is reduced by one if the predicted answer turns out to be incorrect.

By this reward-penalty scheme, statistical information about the learning and the prediction accuracy from all prototypes in F_{2a} to ART_b outputs is built up. This consequently gives a measure of how many times a particular category prototype has accurately participated in the learning and prediction phases.

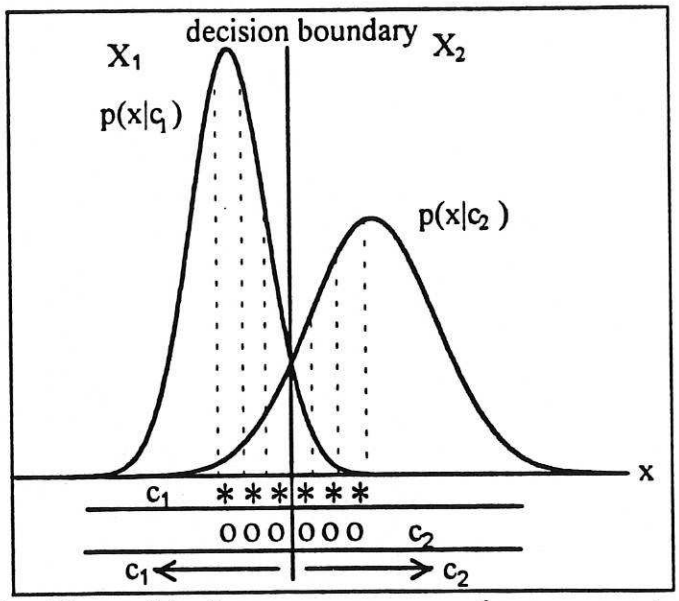

Figure 7 The effects of the frequency measure scheme. c_1 prototypes are denoted by * whereas c_2 prototypes are denoted by o. In the X_1 region, because $p(x|c_1) > p(x|c_2)$, c_1 prototypes should have a higher frequency count than c_2 prototypes and would be selected to predict c_1 as the output. The same happens in X_2. Thus, this looks as if the decision boundary has divided the prototypes into two groups with c_1 prototypes associated with X_1 and c_2 prototypes associated with X_2.

For the Gaussian experiment, as depicted in figure 7, because the conditional probability density $p(x|c_1) > p(x|c_2)$ in X_1, the number of x belonging to c_1 is greater than the number of x belonging to c_2. In other words, more input pairs of x and c_1 occur in the X_1 region. Hence, by the above scheme, those prototypes of c_1 in X_1 should have a higher frequency count than those prototypes of c_2. The same argument

applies to the X_2 region. As a consequence, when $x \in X_1$, c_1 prototypes should be selected and similarly when $x \in X_2$, c_2 prototypes should be selected.

As explained in section 3.2(c), the prototype in F_{2a} is selected in sequence 1,2,... Assume that for a particular sub-range in X_1, a c_2 prototype is established before a c_1 prototype. Then any x falling into that sub-range would generate a predicted output of c_2. With the above scheme, fuzzy ARTMAP is able to select the c_1 prototype based on the frequency count information. This in turn reduces the misclassification rate remarkably, as demonstrated in the experiment.

However, for prototypes adjacent to the decision boundary, the frequency count information may not be reliable because $p(x|c_1)$ is close to $p(x|c_2)$. A large amount of data has, therefore, to be employed in order to obtain accurate frequency information relative to the conditional probability densities.

4 Simulation Studies

4.1 Motivation of Experiment

In statistical decision theory, the general model for pattern recognition is based on the minimisation of misclassification such as the average/expected cost function formulated in Bayes decision criterion (Fu 1982). For a neural network classifier, it is desirable if the network outputs can be treated as estimates of the Bayesian posterior probabilities and thus operate as an optimal Bayes classifier. Besides, interpretation of network outputs as Bayesian posterior probabilities also allows multiple networks to be combined in hierarchy for higher level decision making (Richard & Lippmann 1991).

Unlike some probability-based neural networks (Specht 1990, Musavi *et al* 1993), fuzzy ARTMAP is not designed according to probabilistic arguments. It is not obvious how well fuzzy ARTMAP can perform in classifying statistical data where the class distributions are densely intersected. The task of classifying Gaussian distributed random variables has often been considered as a benchmark problem in statistical pattern recognition. It has been used to analyse the performance of several neural network classifiers (Kohonen *et al* 1988, Yair & Gersho 1990, Richard & Lippmann 1991). In view of this, the experiment of separating Gaussian distributed random variables is chosen. For the sake of simplicity, it is restricted to a single dimensional two-class problem. Thus, it is a binary decision with single observation problem. The difficulty of the task ranges from hard (two densely overlapped classes) to easy (two well-separated classes) by varying the mean value of the two classes. In addition, the effects of different variances and prior probabilities have also been evaluated.

The main purpose of these experiments is to assess the performance of fuzzy ARTMAP as a classifier in classifying statistical data and to examine whether the results from fuzzy ARTMAP and the modified version can approach the Bayes optimal results for a binary decision problem. This work concentrates on the statistical accuracy of the results without taking into account other factors such as network complexity and computational speed.

4.2 Experimental Procedure

Two classes of Gaussian distributed random numbers were generated and normalised between 0 and 1. All input samples were presented in pairs: the random number to ART_a and its target class to ART_b. Complement coding was used. As an example,

Input vector to fuzzy ART_a	=	(x, 1-x)	-- sample x
Input vector to fuzzy ART_b	=	(c, 1-c)	-- target class

Note that class 1 (c_1) was represented by c=0 whereas class 2 (c_2) was represented by c=1. The objective of the experiment was to learn to place an input sample, x, into categories c_1 or c_2 by using the fuzzy ARTMAP on-line classification approach.

Three experiments were carried out:

(1) Two-class classification with different source (mean) separation (equal prior probabilities and fixed variances)

(2) Two-class classification with different prior probabilities (fixed means and equal variances)

(3) Two-class classification with different variances (fixed means and equal prior probabilities)

All experiments employed the single-epoch on-line strategy with fast learning (i.e. learning parameter, $\beta = 1$) (Carpenter *et al*, 1991b). On-line learning has the advantage of imitating the conditions of a human operating in a natural environment. The operational cycle proceeds as follows. First, an input sample x was presented to ART_a. In order to implement a one-to-many mapping as explained in section 3.4, the two nodes with the highest response in F_{2a} were chosen. Based on the frequency information, the one with a higher frequency count was selected as the winner to give a predicted output at ART_b. The output was then compared with x's target class to determine its correctness. This produced a result for the classification accuracy (prediction phase). Learning then ensued to associate the input vector with the target vector (learning phase).

For all experiments, 5000 samples were generated. The first 200 samples were used to "prime" the blank network initially. Performance of the network was evaluated for the remaining 4800 samples. This ensured that early poor predictions did not bias the overall performance unduly.

Fuzzy ARTMAP can be very sensitive to its parameters, especially the vigilance parameters. After some trial-and-error, the network parameters were set to:

ART_a :	Learning parameter	= 1.0 (fast learning)
	Baseline vigilance	= 0.8
	Choice parameter	= 0.01
ART_b :	Learning parameter	= 1.0 (fast learning)
	Vigilance parameter	= 1.0
	Choice parameter	= 0.01
Map Field :	Vigilance parameter	= 0.9

4.3 Experimental Results and Analysis

4.3.1 Experiment I - Two-class Classification with Different Source Separation

This experiment investigated the performance of fuzzy ARTMAP in classifying two Gaussian sources with different source separation. The source separation was defined as the absolute difference between the means of the two classes, i.e., $|\mu_1 - \mu_2|$. The source separation was changed from 0, 1,..., 9 by varying the means away from the y-axis in both positive and negative directions i.e., from two identical sources (mean $\mu_1=\mu_2=0.0$) to two distinct sources ($\mu_1=-4.0$, $\mu_2=4.0$). Hence, the two classes of random numbers were symmetrical to the y-axis. The two sources were equiprobable (prior probability $P\{c_1\}= P\{c_2\}=0.5$) with their variances fixed to 1.0 ($\sigma_1^2 = \sigma_2^2 = 1.0$).

All input values to ART_a were normalised between 0 and 1 by a logistic/sigmoidal function ($1/(1+e^{-x})$). This sigmoidal normalisation technique has the advantage of covering the input range from negative infinity to positive infinity.

4.3.1.1 Experimental Results

Source Separation	Bayes Limit	Fuzzy ARTMAP			Modified Fuzzy ARTMAP		
		Average	ARTa Categories		Average (without freq. measure)	Average (with freq. measure)	ARTa Categories
0	50	50.03	589		50	49.57	620
1	69.15	58.12	499		58.05	66.35	531
2	84.13	73.06	356		73.17	82.42	373
3	93.32	87.79	178		87.87	92.99	189
4	97.72	95.58	77		95.62	97.3	80
5	99.38	98.84	34		98.84	99.25	34
6	99.87	99.71	12		99.71	99.79	12
7	99.98	99.94	8		99.94	99.94	8
8	100	100	4		100	100	4

Table 1 Classification results (average of 5 runs), expressed as percentages, for different source separations

Table 1 summarises the average results of 5 runs. In general, fuzzy ARTMAP and modified fuzzy ARTMAP (without frequency measure) gave very similar results. The greatest difference was only 0.11% for source separation = 2. For comparison, we would concentrate on the performance of fuzzy ARTMAP and modified fuzzy ARTMAP (with frequency measure).

Figure 8(a) depicts the Bayes limit and the average results of fuzzy ARTMAP and modified fuzzy ARTMAP (with frequency measure). We can see that modified fuzzy ARTMAP (with frequency measure) showed an improvement on the classification results. All its results were within 2.8% of the Bayes limit. From table 1 and figure 8(a), the results can be interpreted in 3 groups: source separation = 0 as a special case; source separation = 1, 2, 3, 4 as "hard tasks"; and source separation = 5, 6, 7, 8 as "easy tasks".

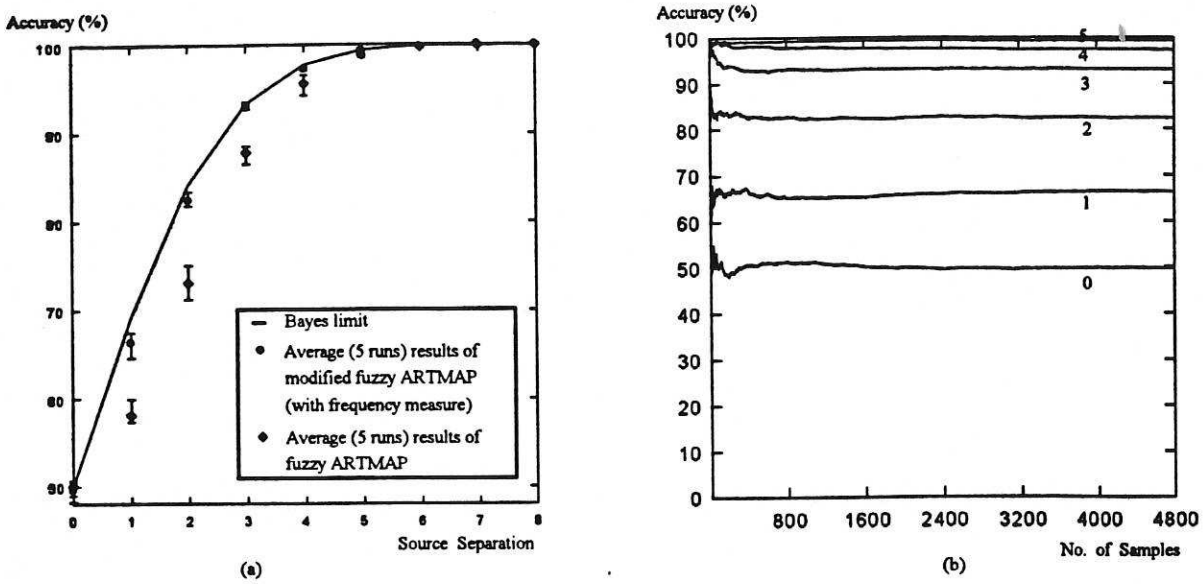

Figure 8 (a) The classification results are bounded by the best and worst results for each source separation. (b) On-line classification accuracy plotted against increasing number of samples and parameterised by source separation.

Source separation = 0 is a special case where all the results were similar to the Bayes limit. This phenomenon will be explained in the discussion. However, for the "hard tasks" a significant improvement was achieved by modified fuzzy ARTMAP (with frequency measure) in approaching the Bayes limits. An increase of 9.36% from fuzzy ARTMAP result was obtained for source separation = 2. For the "easy tasks", both modified fuzzy ARTMAP results and fuzzy ARTMAP results could approach the Bayes limit to within 0.6% since the sources were already well separated.

As expected, for the "hard tasks" modified fuzzy ARTMAP created more categories in ART_a because of the prototype redundancy problem. However, this problem is compensated by the improvements achieved in the results. As a comparison between fuzzy ARTMAP and modified fuzzy ARTMAP (with frequency measure) results, for source separation = 1, there was a 6.41% increase in the number of ART_a categories but the improvement in accuracy was 8.23%. Similarly for source separation = 2, the number of ART_a categories was increased by 4.78% whereas the accuracy was increased by 9.36%.

Figure 8(b) shows a typical on-line classification accuracy plotted against increasing number of input samples. The performance of fuzzy ARTMAP is stable. The results fluctuated at the beginning stage, as expected, and settled to a stable value very rapidly.

4.3.2 Experiment II - Two-class Classification with Different Prior Probabilities

Further tests were carried out to investigate the effects on performance by varying the prior probabilities of the two classes. These experiments were carried out with source separations fixed to 1 (μ_1= -0.5 & μ_2=0.5), 2 (μ_1= -1.0 & μ_2=1.0) and 3 (μ_1= -1.5 & μ_2=1.5). All variances were fixed to 1.0.

For each source separation, the prior probabilities were varied for the following five cases:

(1) $P\{c_1\}=0.1,\ P\{c_2\}=0.9$
(2) $P\{c_1\}=0.2,\ P\{c_2\}=0.8$
(3) $P\{c_1\}=0.3,\ P\{c_2\}=0.7$
(4) $P\{c_1\}=0.4,\ P\{c_2\}=0.6$
(5) $P\{c_1\}=P\{c_2\}=0.5$

4.3.2.1 Experimental Results

P{c1}/P{c2}	Bayes Limit	Fuzzy ARTMAP		Modified Fuzzy ARTMAP		
		Average	ARTa Categories	Average (without freq. measure)	Average (with freq. measure)	ARTa Categories
0.11	90.13	78.08	312	78.2	89.23	336
0.25	81.38	68.32	384	68.52	79.73	410
0.43	74.7	62.6	457	62.68	72.66	500
0.67	70.55	57.9	504	58.05	68.07	534
1	69.15	57.96	495	57.91	66.72	518

(a) Source Separation = 1

P{c1}/P{c2}	Bayes Limit	Fuzzy ARTMAP		Modified Fuzzy ARTMAP		
		Average	ARTa Categories	Average (without freq. measure)	Average (with freq. measure)	ARTa Categories
0.11	93.35	86.05	204	86.18	92.24	215
0.25	88.79	80.09	282	80.36	87.83	295
0.43	86.12	75.95	302	76	84.56	323
0.67	84.62	74.18	343	74.23	83.34	364
1	84.13	72.82	362	72.95	82.48	383

(b) Source Separation = 2

P{c1}/P{c2}	Bayes Limit	Fuzzy ARTMAP		Modified Fuzzy ARTMAP		
		Average	ARTa Categories	Average (without freq. measure)	Average (with freq. measure)	ARTa Categories
0.11	96.63	93.67	125	93.72	96.39	131
0.25	95.02	89.75	156	89.79	94.4	163
0.43	94.04	88.49	179	88.69	93.32	184
0.67	93.49	88.11	165	88.19	93.01	171
1	93.32	87.77	168	88.04	92.9	176

(c) Source Separation = 3

Table 2 Classification results (average of 5 runs), expressed in percentages, for different prior probabilities .

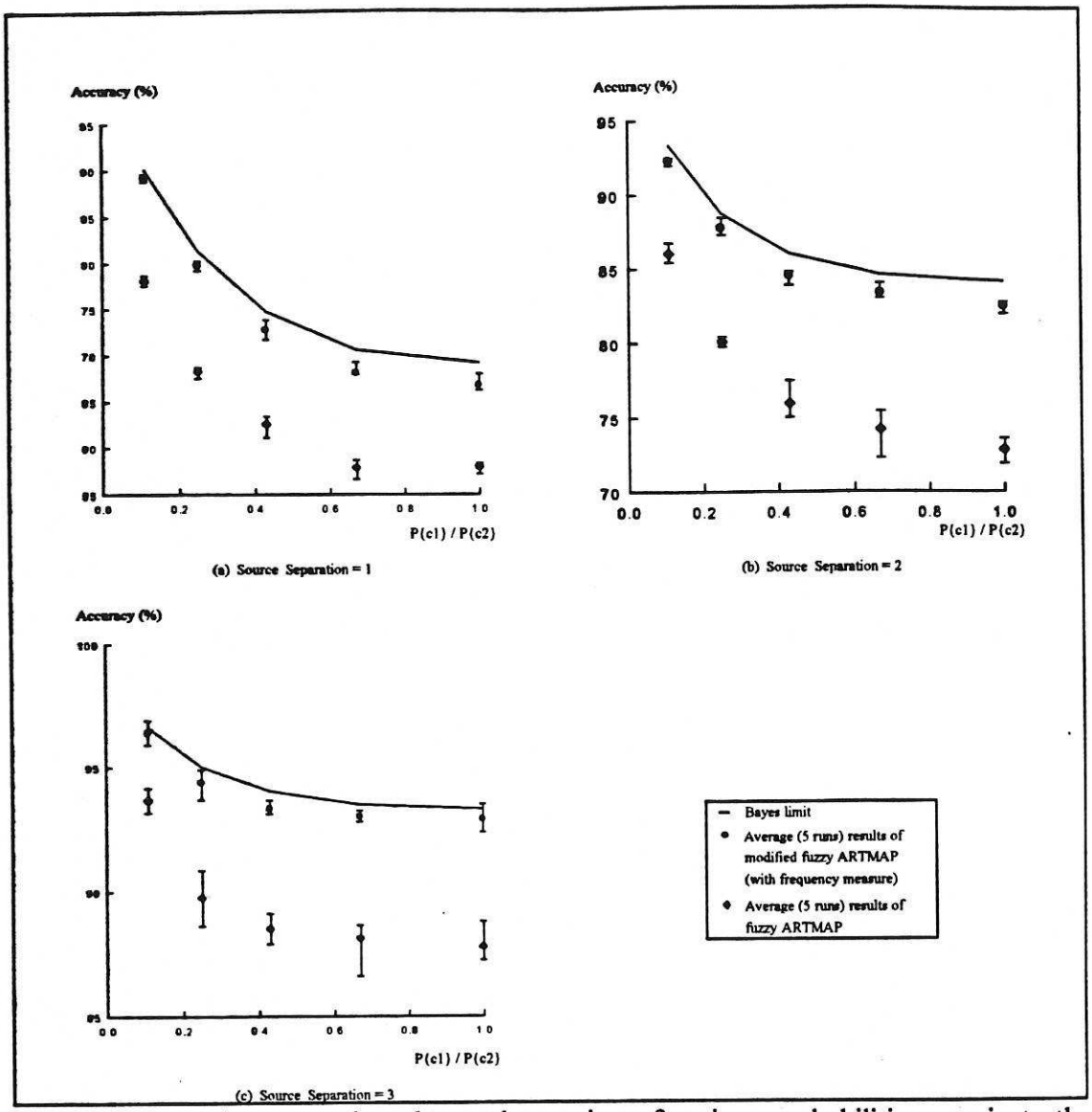

Figure 9 These graphs show the ratio of prior probabilities against the classification accuracy. All the results are bounded by the best and worst results of 5 runs.

Table 2 shows all the classification results. The average results (bounded by the best and worst results) of fuzzy ARTMAP and modified fuzzy ARTMAP (with frequency measure) are shown graphically in figure 9. Again, modified fuzzy ARTMAP (with frequency measure) outperformed fuzzy ARTMAP in all cases. All the results from modified fuzzy ARTMAP (with frequency measure) were within 2.5% of the Bayes limit. Moreover, it showed an improvement in excess of 10% over fuzzy ARTMAP results in some tests for source separation = 1 (a "hard task").

4.3.3 Experiment III - Two-class Classification with Different Variances

In this experiment, the source separation was fixed to 1, 2 and 3 (as in Experiment II) with the prior probabilities $P\{c_1\}=P\{c_2\}=0.5$. The standard deviation of each class was set to:

(1) σ_1=0.5, σ_2=4.5
(2) σ_1=1.0, σ_2=4.0
(3) σ_1=1.5, σ_2=3.5
(4) σ_1=2.0, σ_2=3.0
(5) σ_1=σ_2=2.5

Higher values of standard deviations were used (thus the variances) so that intersection between the two classes could still occur when the source separation was fixed to 1, 2, 3.

4.3.3.1 Experimental Results

All the classification results are tabulated in table 3. Figure 10 depicts the average accuracies plotted against the ratio of standard deviations σ_1/σ_2. As expected, modified fuzzy ARTMAP (with frequency measure) gave a better performance than fuzzy ARTMAP. For instance, for source separation = 1 (a "hard task") an improvement of 9.66% was achieved for σ_1/σ_2 = 0.25. All the results of modified fuzzy ARTMAP (with frequency measure) approximated the Bayes optimal limits to better than 3.6%.

Stdev1/Stdev2	Bayes Limit	Fuzzy ARTMAP		Modified Fuzzy ARTMAP		
		Average	ARTa Categories	Average (without freq. measure)	Average (with freq. measure)	ARTa Categories
0.11	89.26	80.3	251	80.37	88.8	269
0.25	79.17	67.88	437	68.03	77.54	453
0.43	70.18	59.55	503	59.32	67.15	542
0.67	61.89	53.52	536	53.68	58.56	561
1	57.93	51.32	587	51.23	54.41	616

(a) Source Separation = 1

Stdev1/Stdev2	Bayes Limit	Fuzzy ARTMAP		Modified Fuzzy ARTMAP		
		Average	ARTa Categories	Average (without freq. measure)	Average (with freq. measure)	ARTa Categories
0.11	89.87	80.58	259	80.8	88.78	281
0.25	81.06	69.86	368	69.96	78.99	387
0.43	73.04	61.69	461	61.92	70.38	489
0.67	67.16	57.05	531	57.23	64	569
1	65.54	54.31	540	54.48	62.23	562

(b) Source Separation = 2

Stdev1/Stdev2	Bayes Limit	Fuzzy ARTMAP		Modified Fuzzy ARTMAP		
		Average	ARTa Categories	Average (without freq. measure)	Average (with freq. measure)	ARTa Categories
0.11	90.9	82.03	250	82.3	89.12	269
0.25	83.44	72.11	330	72.01	80.99	344
0.43	77.23	65.26	425	65.42	74.66	449
0.67	73.55	61.85	471	61.9	71.48	498
1	72.58	60.46	509	60.68	70.68	558

(c) Source Separation = 3

Table 3 Classification results (average of 5 runs), expressed in percentages, for different variances .

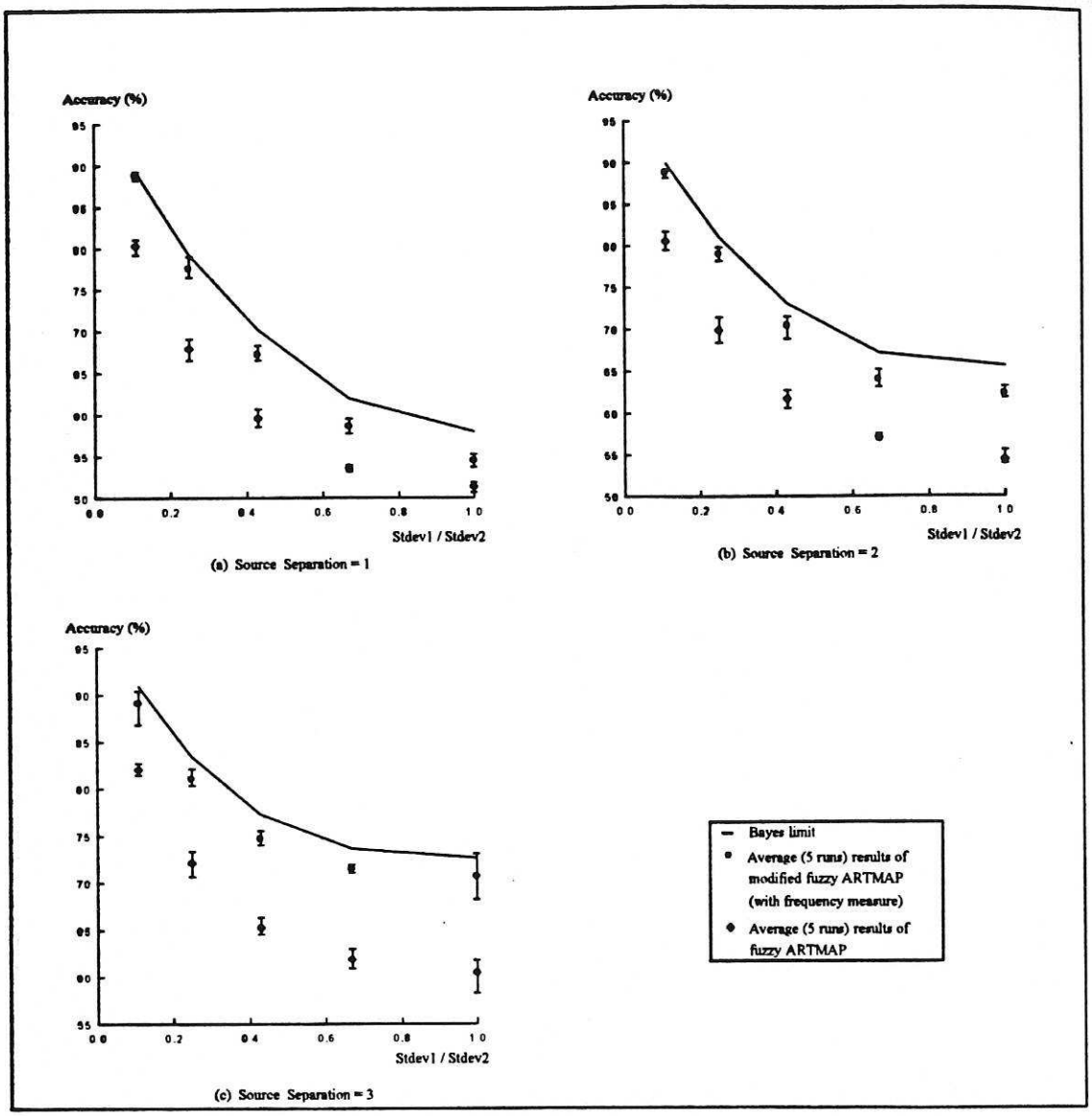

Figure 10 Classification results from differing the ratio of the two variances. The average results of 5 runs are bounded by the best and worst performance.

4.4 Discussion

From Experiment I, several observations can be made. One interesting point is the result for the two identical sources, i.e., source separation = 0. The performance of fuzzy ARTMAP and modified fuzzy ARTMAP (both versions) was close to the Bayes optimal limit of 50% accuracy. The result can be explained by a coin-flipping example. Let us select a prototype of either head or tail. We flip a coin and compare the outcome with our prototype to determine the result. Because the probability of getting a head (Prob(head)) is the same as the probability of getting a tail (Prob(tail)), eventually we should reach an accuracy of 50%. Therefore, when a coin is flipped and the outcome is compared with a prototype, the probability is always approaching 50% regardless either prototype of head or tail is used. The same principle applies here. Referring to figure 11, $p(x|c_1) = p(x|c_2)$ for every input x. For every sub-range, the

likelihood that c_1 generated x and c_2 generated x is the same, just like Prob(head) = Prob(tail). Since it is a 50%-50% case, it is not important which class prototype in F_{2a} is chosen to predict the output. This phenomenon was reflected in the experiment where the original fuzzy ARTMAP result could closely approach the Bayes limit

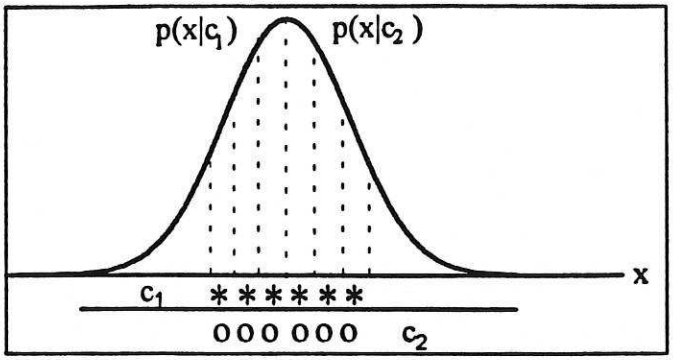

Figure 11 For two identical sources, $p(x|c_1) = p(x|c_2)$ for the entire input range. Prototype selection becomes insignificant because the probability of x belonging to c_1 is the same as the probability of x belonging to c_2.

However, prototype selection becomes critical for separated cases because their conditional probability densities are no longer identical. This could be observed from those "hard tasks" (source separation = 1, 2, 3 & 4) where the results of fuzzy ARTMAP degraded seriously. For source separations of 1 and 2, the results were 11% away from the Bayes limit. This problem was overcome by the frequency measure approach. Because $p(x|c_1) > p(x|c_2)$ for $x \in X_1$ (figure 7), this information should be reflected in the frequency count which in turn enabled c_1 prototypes to be chosen. The same argument applies for $x \in X_2$. When "similar prototypes" (which were linked to different outputs) occurred in X_2, modified fuzzy ARTMAP (with frequency measure) would select the one with a higher frequency count and hence minimise the misclassification rate. An improvement of 8-9% in accuracy was shown for source separations of 1 and 2.

For the "easy tasks" (source separation = 5, 6, 7 & 8), all the results were able to approach the Bayes limit to within 1% accuracy. When the sources are well-separated, their distributions do not densely overlap each other. Thus, modified fuzzy ARTMAP (both versions) and fuzzy ARTMAP could accurately classify almost all input samples by establishing appropriate prototypes which reflected their respective conditional probability densities.

A different learning strategy has also been investigated. We experimented on the "hard tasks" using the fast-commit slow-recode learning rule (Carpenter *et al*, 1992). All other parameters were as given in section 4.2 except that the learning parameter, β, was set to 0.5 for committed nodes. However, very similar classification results as those from the fast-learning cases (Experiment I) were obtained. As an instance, for source separation = 2, the average (5 runs) accuracy of fuzzy ARTMAP was 72.65%; whereas the average (5 runs) accuracy of fuzzy ARTMAP (with frequency measure) was 82.27%. The absolute differences for all the average results were within 1% of the results from Experiment I.

To evaluate further the classification ability of modified fuzzy ARTMAP (with frequency measure) in producing outputs which approximate the Bayes optimal results, various configurations of the Gaussian problem have been examined. Experiment II investigated the effects of differing the prior probabilities of the two classes; whereas Experiment III investigated the effects of differing their variances. Again, all the results were able to approach the Bayes limit to within 3.6% at worst.

By inspecting the modified fuzzy ARTMAP (with frequency measure) results, we notice that very infrequently individual run results exceed the Bayes optimal limit. This could be due to the prototypes adjacent to the decision boundary of the two classes. As explained in section 3.6, because $p(x|c_1)$ is close to $p(x|c_2)$ the frequency count information may not reflect the actual conditional probability densities of the two classes. A lot of data should be used in order to give a higher frequency count for c_1 prototypes in X_1 and vice versa.

As explained earlier, modified fuzzy ARTMAP creates redundant prototypes in ART_a as a side-effect of the implementation of the one-to-many mapping. The number of nodes in F_{2a} was increased, especially for the "hard tasks". However, the improvements achieved in the classification results mitigated this effect. Nevertheless, such redundant prototypes should be eliminated or reduced. One suggestion is to incorporate a forgetting factor or a decaying function to each F_{2a} node. Then, rarely-activated or spurious prototypes would gradually fade away until they re-enter the pool of uncommitted nodes. An alternative is to transfer the frequency count information from the F_{2a} layer to the map field so that each ART_a and ART_b module could maintain its original clustering algorithm. These suggestions are the subject of further work.

5 Conclusion

This work investigates the performance of fuzzy ARTMAP in classifying statistical data autonomously and on-line. It highlights the problem of one-to-many mapping in fuzzy ARTMAP. A modification to the fuzzy ARTMAP algorithm is proposed to enable the implementation of one-to-many mapping. In addition, a frequency measure scheme is also suggested to relate how many times a particular category prototype has accurately predicted an output in the on-line operation. The modified algorithm with a frequency measure has empirically shown a significant improvement over the original algorithm in approaching the Bayes optimal classification rates for a binary decision problem, i.e., separating two classes of Gaussian distributed continuous-valued random variables. As demonstrated in the experiments, various configurations of the problem have been investigated by varying the source separation, prior probabilities and variances of the two classes. In all cases, modified fuzzy ARTMAP with a frequency measure outperformed fuzzy ARTMAP to better approach the Bayes optimal results.

Acknowledgements

C. P. Lim gratefully acknowledges the financial support of the Committee of Vice-Chancellors and Principals (CVCP), UK and the Department of Automatic Control and Systems Engineering, University of Sheffield, UK.

REFERENCES

Burke, L.I. (1991). Clustering Characterization of Adaptive Resonance. *Neural Networks*, 4, pp 485-491.

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

Carpenter, G.A. (1987b). ART 2: Stable Self-Organizing of Pattern Recognition Codes for Analog Input Patterns. *Applied Optics*, 26, pp. 4919-4930.

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

Carpenter, G.A., Grossberg, S., (1990). ART 3: Hierarchical Search using Chemical Transmitters in Self-Organizing Pattern Recognition Architectures. *Neural Networks*, 3, pp. 129-152.

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

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

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

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

Dunteman, G.H. (1984). *Introduction to Multivariate Analysis*. Beverly Hills, USA: Sage Publications.

Fu, K.S. (1982). *Applications of Pattern Recognition*. Boca Raton, USA: CRC Press.

Hrycej, T. (1992) Loss Function Based Neural Classifiers. *Proc. of Int. Conf. on Artificial Neural Networks (ICANN-92)*, Brighton. 2, pp 1135-1138.

Kohonen,T., Barna, G., Chirsley, R. (1988). Statistical Pattern Recognition with Neural Networks: Benchmarking Studies. *Proc. IEEE Int. Conf. on Neural Networks*, San Diego. pp I61-68.

Light, W.A. (1992) Some Aspects of Radial Basis Function Approximation. *Approximation Theory, Spline Functions and Applications*. 356, pp 163-190.

Melsa,J.L., Cohn,D.L. (1978). *Decision and Estimation Theory*. Tokyo:McGraw-Hill.

Moore, B. (1989). ART1 and Pattern Clustering. In Touretzky, D., Hinton, G., & Sejnowski, T. (Eds.) *Proc. 1988 Connectionist Models Summer School*. pp 174-185. San Mateo, CA: Morgan Kaufmann Publishers.

Musavi, M.T., Kalantri, K., Ahmed, W., Chan, K.H. (1993) A Minimum Error Neural Network (MNN). *Neural Networks*, 6, pp 397-407.

Poggio, T., Girosi, F. (1990) Network Approximation and Learning, *Proceeding of IEEE*, 78, 9, pp 1481-1497.

Richard,M.D., Lippmann,R.P. (1991). Neural Network Classifiers Estimate Bayesian a posteriori Probabilities. *Neural Computation*, 3, pp 461-483.

Ryan, T.W. (1988). The Resonance Correlation Network. *Proc. IEEE Int. Conf. on Neural Networks*, San Diego. pp I673-680.

Specht, D.F. (1990) Probabilistic Neural Networks. *Neural Networks*, 3, pp 109-118.

Wan, E.A. (1990). Neural Network Classification: A Bayesian Interpretation. *IEEE Trans. on Neural Networks*, 1(4), pp 303-305.

White, H. (1989). Learning in Artificial Neural Networks: A Statistical Perspective. *Neural Computation*, 1, pp 425-464.

Yair, E., Gersho, A. (1990). The Boltzmann Perceptron Network: A Soft Classifier. *Neural Networks*, 3, pp 203-221.

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